

MEDIUM ACCESS CONTROL PROTOCOLS WITH FAST COLLISION
RESOLUTION FOR WIRELESS LOCAL AREA NETWORKS

By

YOUNGGOO KWON

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
UNIVERSITY OF FLORIDA

2002

Copyright 2002

by

Younggoo Kwon

Dedicated to my family, who have provided me with support emotionally and financially throughout this long journey ...

ACKNOWLEDGMENTS

I would like to express my gratitude to all those who helped me in completing this dissertation. I am deeply indebted to my advisor Professor Yuguang Fang for his stimulating discussions and encouragements with love as in a family, which helped me to carry out the research in this dissertation. I also thank my co-advisor, Professor Haniph Latchman, and the other committee members Professor Tan Wong and Professor Max Shen, who have provided me with insightful suggestions, which greatly improved the quality of this dissertation. My lovely colleagues in the Wireless Network Laboratory (WINET) are like brothers and sisters, who have supported me throughout my research work. I really want to express my appreciation for their help, support, interest and invaluable hints. Particularly, I am obliged to Wenjing Lou, Wenchao Ma, Yu Zeng, Wei Liu, Xiang Chen and Byungseo Kim. Finally, I would express my special appreciation to my wife, Seunghee, and my son, Kangmo, whose patience, love and support made this dissertation possible.

TABLE OF CONTENTS

	<u>page</u>
ACKNOWLEDGMENTS	iv
LIST OF TABLES	vii
LIST OF FIGURES	viii
ABSTRACT	xi
CHAPTERS	
1 INTRODUCTION	1
1.1 Wireless Local Area Networks	1
1.1.1 Wireless Medium Characteristics	2
1.1.2 Modulation Techniques for High-Speed Wireless Net- works	4
1.2 Medium Access Control Algorithms in WLANs	7
1.3 Fair Scheduling	10
1.4 Quality of Service	12
2 Fast Collision Resolution (FCR) Algorithms	15
2.1 Distributed Contention-Based MAC Algorithms	15
2.1.1 ALOHA and Slotted ALOHA	15
2.1.2 IEEE 802.11 standard Medium Access Control	16
2.1.3 Dynamic Tuning Backoff	20
2.2 Fast Collision Resolution (FCR) MAC Algorithm	23
2.2.1 The Basic Idea	23
2.2.2 Fast Collision Resolution (FCR) Algorithm	26
2.2.3 Performance Analysis of FCR	32
2.2.4 Performance Results for IEEE 802.11 FHSS: 2 Mbps	36
2.2.5 Performance Results for IEEE 802.11 DSSS: 2 Mbps	43
2.2.6 Performance Results for IEEE 802.11b DSSS: 11 Mbps	45
2.3 Effects on Performance at Transport Layer	51
2.3.1 Overview of Transport Layer Protocols	51
2.3.2 Transmission Control Protocol (TCP)	52
2.3.3 User datagram protocol (UDP)	54
2.3.4 Performance Analysis	55

3	FAIRLY SCHEDULED Fast Collision Resolution (FS-FCR) Algorithm	60
3.1	Fair Scheduling Algorithms	60
3.1.1	Generalized Processor Sharing (GPS)	61
3.1.2	Self-Clocked Fair Queueing (SCFQ)	62
3.1.3	Distributed Self-Clocked Fair Queueing	63
3.2	Fairly-Scheduled Fast Collision Resolution (FS-FCR)	66
3.3	Performance Evaluation	68
4	QOS-based Medium Access Control Protocols	74
4.1	Current QoS-based MAC Algorithms for Real-Time Services	75
4.1.1	Differentiation Mechanism for IEEE 802.11	76
4.1.2	Black Burst Contention	77
4.2	Real-Time Fast Collision Resolution (RT-FCR)	79
4.3	Performance Evaluation	83
4.3.1	Source Models	83
4.3.2	Simulation Results	84
5	CONCLUSIONs and Future Research Directions	89
	REFERENCES	92
	BIOGRAPHICAL SKETCH	101

LIST OF TABLES

<u>Table</u>	<u>page</u>
2.1 Example of IEEE 802.11 MAC with binary exponential backoff	30
2.2 Example of Fast Collision Resolution Algorithm	31
2.3 Network Configurations: IEEE 802.11 FHSS 2Mbps	37
2.4 Throughput Results for FCR Algorithm	37
2.5 Throughput Results for IEEE 802.11 MAC Algorithm	37
2.6 Network Configurations: IEEE 802.11 DSSS 2Mbps	43
2.7 Network Configurations: IEEE 802.11 DSSS 2, 11Mbps	47
4.1 Assigning Backoff Range	80

LIST OF FIGURES

<u>Figure</u>	<u>page</u>
1.1 Wireless Network Architecture	2
1.2 DSSS and FHSS Examples	5
1.3 Examples of OFDM Symbol Subcarriers	6
1.4 OSI Reference Model	7
2.1 Inter Frame Spaces in IEEE 802.11 MAC	17
2.2 Basic operations of CSMA/CA	17
2.3 Random Backoff Procedure	18
2.4 Binary Exponential Backoff	19
2.5 Access Scheme	20
2.6 t_v function for different M values	22
2.7 Distribution for contention window size of sending a packet	35
2.8 Throughput for 10 BE data stations wireless LAN	38
2.9 Throughput for 50 BE data stations wireless LAN	39
2.10 Throughput for 100 BE data stations wireless LAN	39
2.11 Throughput vs. offered load	40
2.12 Delay distribution for 10 stations wireless LAN	41
2.13 Delay distribution for 100 stations wireless LAN	42
2.14 Throughput for 10 BE data stations wireless LAN	43
2.15 Throughput for 50 BE data stations wireless LAN	44
2.16 Throughput for 100 BE data stations wireless LAN	44
2.17 Throughput vs. offered load	45
2.18 Delay distribution for 10 stations wireless LAN	46

2.19	Delay distribution for 100 stations wireless LAN	46
2.20	Throughput for 10 BE data stations wireless LAN	47
2.21	Throughput for 50 BE data stations wireless LAN	48
2.22	Throughput for 100 BE data stations wireless LAN	48
2.23	Throughput vs. offered load	49
2.24	Average Delay	49
2.25	Delay distribution for 10 stations wireless LAN	50
2.26	Delay distribution for 100 stations wireless LAN	50
2.27	Throughput result for FTP traffic sources	55
2.28	Fairness Index for FTP traffic sources	56
2.29	Throughput result for bursty CBR traffic sources	56
2.30	Fairness Index for bursty CBR traffic sources	57
2.31	Throughput result for voice traffic sources	57
2.32	Fairness Index for voice traffic sources	58
2.33	Packet Delivery Ratio	58
3.1	Fair scheduling system model	61
3.2	Fair index for 10 sec simulation	69
3.3	Fair index for 100 sec simulation	69
3.4	Throughput for 10 BE data station wireless LAN	71
3.5	Throughput for 100 BE data stations wireless LAN	71
3.6	Delay Distribution for 10 stations wireless LAN	72
3.7	Delay Distribution for 100 stations wireless LAN	73
4.1	BB Contention Channel Access Scheme	77
4.2	RT-FCR Medium Access Scheme	79
4.3	Priority Scheme of RT-FCR Algorithm	81
4.4	Ratio of Dropped Voice Packets	84
4.5	Throughput of Best-Effort Data Traffic Transmission	85

4.6	Ratio of Dropped Real-Time Packets vs. Number of CBR Stations . .	86
4.7	Throughput of Best-Effort Data Traffic vs. Number of CBR Stations	86
4.8	Ratio of Dropped Real-Time Packets vs. Number of VBR Stations . .	87
4.9	Throughput of Best-Effort Data Traffic vs. Number of VBR Stations	87

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

MEDIUM ACCESS CONTROL PROTOCOLS WITH FAST COLLISION
RESOLUTION FOR WIRELESS LOCAL AREA NETWORKS

By

Younggoo Kwon

August 2002

Chair: Yuguang Fang

Major Department: Electrical and Computer Engineering

The development of efficient medium access control (MAC) protocols, which provide both high throughput performance for data traffic and good quality of service (QoS) support for real-time traffic, is the current major focus in wireless medium access control (MAC) research. This dissertation focuses on the distributed contention-based MAC protocols based on the carrier sense multiple access (CSMA) scheme, targeted at improving throughput, maintaining a high degree of fairness for serving users, and providing QoS for real-time services. To provide all the required properties for MAC protocols in wireless networks, we propose an efficient contention resolution algorithm for wireless local area networks, namely the *Fast Collision Resolution* (FCR) algorithm. The MAC protocol with this new algorithm attempts to provide significantly high throughput performance for data services while maintaining the simplicity of implementation. The FCR algorithm is compared with the IEEE 802.11 MAC and is shown that higher throughput can

be achieved. To provide fairness for serving users, the distributed self-clocked fair queueing (SCFQ) algorithm is modified and incorporated into the FCR algorithm with a maximum successive transmission limit controlled by the SCFQ algorithm, resulting in a new protocol called *Fairly Scheduled FCR* (FS-FCR). To provide the QoS support in the MAC layer, we apply the priority scheme based on service differentiations for real-time services in FCR algorithm and develop the new MAC protocol called *Real-Time FCR* (RT-FCR). Extensive simulation studies have been carried out to evaluate the FCR, the FS-FCR, and the RT-FCR and show that the FCR improves the throughput performance significantly, the FS-FCR provides a high degree of fairness while maintaining the high throughput performance of the FCR algorithm, and the RT-FCR supports the desired QoS for voice, video, and data services.

CHAPTER 1 INTRODUCTION

Wireless networking has been the subject of intensive study and recent advancement largely due to its convenience for mobility and the advantage of no wires. Wireless communications enable untethered communication with anyone, anywhere, and anytime. Recently, high-speed wireless data communication systems have been installed and operated in many places around the world. The wireless medium is a shared medium, and therefore multiple users may attempt to access the medium at the same time. Multiple transmissions at the same time may result in corrupted data which make communication difficult. A medium access control (MAC) protocol efficiently controls the access to the shared wireless medium by regulating the packet transmissions of all users, thus allowing each user to communicate with each other efficiently. Medium access control protocols for the wireless medium have been proposed mainly to improve network throughput; however, recently other considerations such as fairness, quality of service (QoS), and multi-layer optimizations have been included in new wireless MAC protocol design.

1.1 Wireless Local Area Networks

Wireless local area networks, comprised of devices such as access points (APs) and mobile stations, use high-frequency electromagnetic waves to transmit information from one station to another. An access point is a special device in the network that connects to the fixed infrastructure and provides information access to other data networks. Wireless network architectures can be categorized into two classes: distributed and centralized. Figure 1.1 shows the typical wireless local area

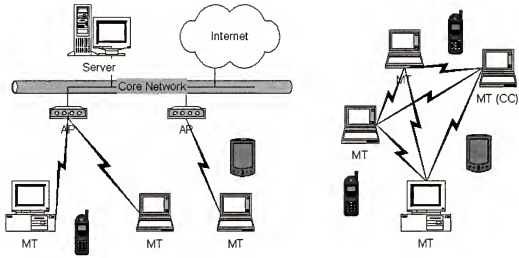


Figure 1.1: Wireless Network Architecture

network architecture consisting of both distributed and centralized wireless network segments.

Distributed wireless networks are networks in which wireless stations communicate with one another without any centralized infrastructured devices. Wireless stations have a wireless interface and exchange information between one another in a distributed manner. Centralized wireless networks feature an infrastructure controller or scheduler which acts as the interface between wireless and wireline networks and also regulates the transmission rules among wireless mobile stations. Centralized wireless networks provide a high degree of flexibility in the design of MAC algorithms because various scheduling or controlling transmission algorithms can be implemented in the access point to satisfy different QoS requirements. Generally, wireless local area networks should support these two different network architectures.

1.1.1 Wireless Medium Characteristics

Wireless communications have special properties such as broadcasting, contention and the limited channel capacity due to interference, fading and

multipath effects. Therefore, the wireless medium exhibits the characteristics of a “half-duplex” operation, time varying channel, with bursty channel error, and location dependent carrier sensing. These properties make the design of MAC protocols much more difficult comparing to wireline networks. Because of the time-varying channel and varying signal strength, high transmission error rate is expected in wireless communications. In wireline networks, the bit error rate is typically less than 10^{-6} and as a result the probability of a packet error is small. In contrast, wireless channels may have bit-error rates as high as 10^{-2} , resulting in a much higher probability of packet errors.

In free space, signal strength attenuates proportional to the square of the distance between transmitter and receiver. As a result, carrier sensing is a function of the position of the receiver relative to the transmitter. Due to various fading and unknown interference and the sensing circuit design in wireless devices, collision detection is much more difficult. Therefore, many collision avoidance schemes are proposed in wireless networks instead of using the collision detection scheme like in Ethernet. Also, the location dependent carrier sensing results in hidden nodes and exposed nodes. A hidden node is one that is within the range of the receiver but out of range of the transmitter. An exposed node is complementary to a hidden node, namely, it is one that is within the range of the transmitter but out of range of the receiver. A hidden node may cause a collision at the receiver, which degrades the throughput. An exposed node cannot transmit due to the transmission of the transmitter although it would have been perfectly okay to transmit as long as its intended receiver is not in the range of the transmitter. Thus, both hidden nodes and exposed nodes may cause degradation of the collision avoidance based MAC protocols if not appropriately addressed.

1.1.2 Modulation Techniques for High-Speed Wireless Networks

Modulation is the process of encoding information from a message source in a manner suitable for the transmission over a specific medium. The modulation schemes used in wireless communications can be categorized as either narrowband modulation or wideband modulation. Spread-spectrum techniques account for one type of wideband modulation which are used in wireless communications. OFDM is another wideband technique that is used for high-speed wireless communications. In this section, we provide basic descriptions and characteristics of modulation schemes (narrowband modulation, spread spectrum, orthogonal frequency division multiplexing) that are appropriate for use in wireless medium.

Narrowband modulation is usually achieved by modulating some combination of the amplitude, phase, or frequency of a carrier waveform. Modulations that rely on the amplitude to carry information, such as amplitude-shift keying (ASK) and quadrature amplitude modulation (QAM), require accurate estimates of the channel gain. In general amplitude modulations result in the greatest bandwidth efficiency but require the greatest channel knowledge. Modulations that rely on the phase to carry information are less affected by channel amplitude variations. Frequency modulation is relatively insensitive to amplitude and phase variations, but typically has the lowest bandwidth efficiency.

Spread-spectrum transmission is a method that spreads the transmitted signal bandwidth much wider than is necessary to send the information. Spread-spectrum signals are generally used to overcome the harmful effects of narrowband interference due to jamming, interference arising from other users of the channel, and self-interference due to multipath propagation. Spread-spectrum signals can be hidden by transmitting at low power level at any given narrow frequency band, thus making it difficult for an unintended listener to detect the signal in the presence of background noise. Code-division multiple access (CDMA), one

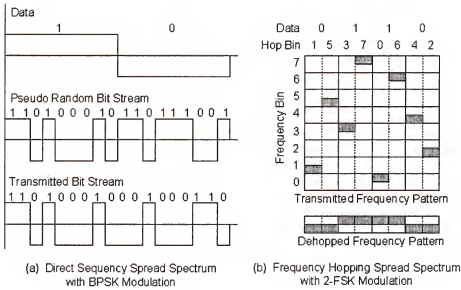


Figure 1.2: DSSS and FHSS Examples

of the commonly used spread-spectrum transmission techniques, allows multiple users to simultaneously use a common channel for transmissions of information. The processing gain, the ratio of the bandwidth of the transmitted signal (i.e., the spread bandwidth) to the information bandwidth, is a measure of the performance.

The most common spread-spectrum technologies are direct-sequence spread spectrum (DSSS) and frequency-hopping spread spectrum (FHSS). DSSS modulates the data signal by a high rate pseudo-random sequence of phase-modulated pulses before shifting the signal to the carrier frequency band for transmission. FHSS spreads the transmission spectrum of a data signal by randomly hopping over different carrier frequencies. In Figure 1.2, basic examples of DSSS and FHSS are shown. Since spread spectrum provides a good solution for overcoming many problems of wireless channels, it is widely used for high-speed wireless local-area networks.

The OFDM technique has a great potential for coping with the shortcomings of broadband wireless communications. OFDM is a special case of multi-carrier transmission, in which a single data stream is transmitted over a number of

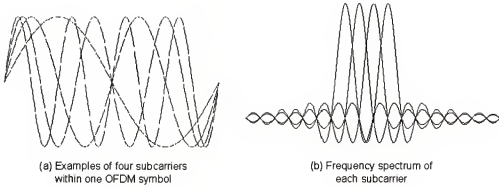


Figure 1.3: Examples of OFDM Symbol Subcarriers

lower-rate subcarriers[80]. It can be seen as either a modulation technique or a multiplexing technique. In a single carrier system, a single fade or interference can cause the entire link to fail, but in a multicarrier system, only a small percentage of the subcarriers will be affected. The carriers in an OFDM signal are overlapped orthogonally, so that the overlapped sidebands of the individual carriers do not affect each other, as shown in Figure 1.3. At the center frequency of each subcarrier in OFDM signals, there is no crosstalk from other channels because of the orthogonality among subcarriers. Therefore, at the receiver, the transmitted data are recovered by calculating the correlation values at the center frequency of each subcarrier. OFDM is an efficient way to deal with multipath problems in communication channels. In relatively slow time-varying channels, it is possible to significantly enhance the capacity by adapting the data rate per subcarrier according to its SNR (bit-loading). OFDM is robust against narrowband interference because such an interference affects only a small percentage of the subcarriers.

OFDM is typically used with a cyclic prefix or guard interval to avoid intersymbol interference, but this comes at the expense of the data rate. In some applications, equalization may be used to reduce the size of the guard interval, but this typically requires a complicated interleaver. The OFDM technique alleviates time delay spread and is adaptable to large narrowband interference and frequency

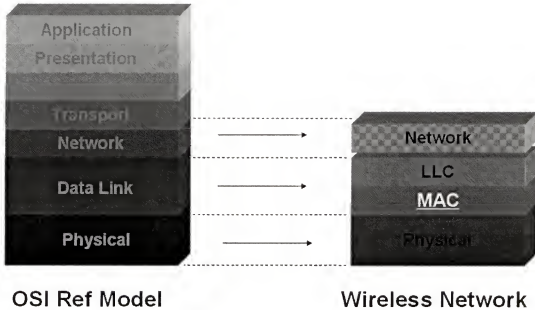


Figure 1.4: OSI Reference Model

nulls. Considering the high degree of flexibility for handling many problems over wireless communications, OFDM seems to be a promising candidate for high-speed wireless communications.

1.2 Medium Access Control Algorithms in WLANs

A good medium access control (MAC) algorithm for wireless LANs should provide an efficient way to share limited channel resources, together with simplicity in operation, fairness for serving all stations, and high throughput. It should give low delay in low network load situations, and high throughput under high network load conditions, although it is usually difficult to satisfy both. Most medium access control algorithms in wireless LANs can be divided into two broad categories, namely contention-based medium access control algorithms and reservation-based medium access control algorithms. The contention-based medium access control algorithms are generally used in distributed network architectures, suitable for bursty data traffic under low network load because of their low

delay characteristics and the simplicity of implementation, which also gives good solutions for the problems of ad hoc wireless networks where no infrastructure access point exists[12, 18, 53]. The reservation-based medium access control algorithms such as guaranteed access protocols and hybrid access protocols (i.e., distributed or random access reservation-based medium access control algorithms) are used by an access point in centralized network architectures. Reservation-based medium access control algorithms can easily support the required QoS for each traffic type and work efficiently under heavy network load conditions. However, reservation-based medium access control algorithms suffer from complex system architectures, and huge overheads under low network load conditions and various user populations[23, 45]. In a guaranteed access protocol, stations access the medium in a round-robin way. There are two ways to implement these protocols. One is to use a master-slave configuration such as a polling system. The second is to operate in a distributed manner by exchanging tokens such as a token-passing system. Hybrid access protocols combines contention-based and reservation-based protocols to design more efficient MAC protocols. Most hybrid access protocols are based on request-grant mechanisms. Each station sends a request to the base station by using a contention-based access protocol. The base station then schedule the transmission order and sends a grant to the station for data transmissions.

In a contention-based random access protocol, stations contend for access to the medium, so if only one station transmits its packet, the packet is delivered successfully. If multiple stations transmit at the same time, then a collision occurs. To resolve the collisions, various kinds of MAC algorithms have been proposed. ALOHA was the first protocol, and it is operated in a completely distributed manner with simple operations. However, it does not use the carrier sensing mechanism, and results in poor throughput performance. CSMA is one of the most pervasive MAC schemes, and is a simple distributed protocol whereby stations

regulate their packet transmission attempts based solely on the current shared channel status. In most schemes similar to CSMA, each station that participates in a collision schedule the retransmission of its packet after a random period of time, in the hope of avoiding another collision.

Multiple access with collision avoidance (MACA) uses a three-way handshake as a solution to the hidden node problem. A station that has data to send transmits a short request to send (RTS) packet. All stations within one hop of the transmitting station hear the RTS and defer their transmissions. The destination, upon successfully receiving the RTS, responds with a clear to send (CTS) short packet. All stations within one hop of the destination station hear the CTS and will defer their transmissions. On receiving the CTS, the transmitting station assumes that the channel is acquired and initiates the data transmission. This handshaking mechanism does not completely solve the hidden terminal problem, but it does prevent it to a large extent. In environments without hidden nodes, MACA may improve the throughput of the network over that of CSMA because collisions involve only short RTS packets rather than normal data packets as in CSMA. The Floor Acquisition Multiple Access (FAMA) class of protocols include several variants of MACA algorithm. Another variation of MACA is the Distributed Foundation Wireless MAC (DFWMAC) which has developed into the basic access protocol in the IEEE 802.11 standard.

The Elimination Yield-Non Preemptive Priority Multiple Access (EY-NPMA) is the channel access protocol used in the HIPERLAN system being developed in Europe. The protocol operates as follows: A station that has data to transmit senses the medium for a period corresponding to the time it takes to transmit 1700 bits. If no transmission is heard the channel is considered idle and the station can start transmitting its packet immediately. If the channel is busy, the station synchronizes itself at the end of the current transmission interval and

contends for the channel. The channel access has three phases: prioritization phase, contention phase, and transmission phase. The contention phase consists of two sub-phases: elimination phase and yield phase. In the elimination phase each station transmits for a random number of slots. At the end of the elimination phase, the station turns around and listens to the channel. If the channel is busy, it aborts its transmission attempt. If the channel is idle the station moves to the yield phase. In this phase, it listens to the channel for a random number of slots. If no transmission is detected during this time, the station starts and completes its data transmission.

However, it is well known that if the number of users and network load increase, the performance of the distributed contention-based MAC algorithm degrades significantly because of the excessively high collision rate. Many researchers have focused on analyzing and improving the performance of the distributed contention-based MAC algorithm[14, 18]. To increase the performance, an efficient collision avoidance technique which can reduce the wasting overheads for each contention procedure is needed. To this end, many novel channel access algorithms have been proposed, such as improved backoff algorithms which change the increasing and decreasing factor, others which change the contention window size and the random backoff values, or use out-band busy-tone signal and append the contention information on the transmitted packets[12, 13, 33, 38].

1.3 Fair Scheduling

Fairness is another important issue in MAC protocol design for wireless local area networks. Many wireless data networks try to support various applications such as multimedia teleconferencing, WWW browsing, etc. Supporting such applications requires the network to provide quality of service as well as fairly sharing limited wireless networking resources. In wireline networks, these requirements are typically satisfied by a resource reservation or fair scheduling algorithms. However,

in wireless networks, new factors such as the mobility and channel error make it very difficult to perform either resource reservation or fair scheduling. While there have been some recent efforts to provide resource reservation and fair scheduling in wireless data networks, the problem has remained largely unaddressed.

The design of a scheduling algorithm should consider delay, complexity of implementation, and fairness factors. A large delay bound implies increased burstiness of the session, thus increasing the amount of buffering needed in the switches to avoid packet losses. The delay behavior generally requires insensitivity to traffic patterns, delay bounds that are independent of the number of stations, and the ability to control the delay bound. Most fair scheduling algorithms have been studied under the assumption that a centralized scheduler (or server) allocates the limited resources fairly among flows based on various factors such as arrival rates, delay constraints and bandwidth requests. The well-known generalized processor sharing (GPS) algorithm is based on a fluid-flow model, which is generally regarded as an idealized fair scheduling algorithm. GPS has been proven to have two important properties: 1. It can provide an end-to-end bounded delay service to a leaky-bucket constrained session. 2. It can ensure fair allocation of bandwidth among all backlogged sessions regardless of whether or not their traffic is constrained. The former property is the basis for supporting guaranteed services while the latter property is important for supporting best-effort and link-sharing services. While GPS is a fluid model that cannot be implemented in practice, various packet approximation algorithms are designed to provide services that are almost identical to that of GPS.

The packet by packet version of GPS (PGPS), and the weighted fair queueing (WFQ) are the simple examples of variations of the GPS system. A GPS system is simulated in parallel with the packet by packet system in order to identify the set of connections that are backlogged at each instant. A timestamp for each arriving

packet, indicating the time at which it would depart the system under GPS, is calculated. Packets are then transmitted in increasing order of their timestamps. A serious problem with this approach is its computational complexity. In order to reduce its complexity, an approximate implementation of GPS multiplexing was proposed by Davin and Heybey and later analyzed by Golestani under the name self clocked fair queueing (SCFQ)[43]. In this implementation, the timestamp of an arriving packet is computed based on the timestamp of the packet currently in service. This approach reduces the complexity of the algorithm greatly. The virtual clock scheduling algorithm, on the other hand, provides the same end to end delay and burstiness bounds as WFQ with a simple timestamp computation algorithm.

Fair scheduling issues in wireless local area networks have different characteristics from the traditional fair scheduling in wired networks. In many wireless LANs, such as the IEEE 802.11 LANs, the dominant mode commonly used in practice is the DCF mode, which is operated based on the distributed contention-based MAC protocols. There is no central controller to assign the fair scheduler such as GPS or its many variants. Therefore, we need to consider carefully the use of distributed fair scheduling in wireless LANs[98]. It is noted that the IEEE 802.11 MAC has inherent unfairness characteristics[62, 89, 98]. In the IEEE 802.11 MAC protocol, the station, which has succeeded in packet transmission, will set its contention window size to its minimum allowable value. This implies that the station with a successful packet transmission will have higher probability of gaining access of the medium and succeed again in the next contention period, leading to unfairness.

1.4 Quality of Service

A very significant paradigm shift in communications networks is the convergence of voice, video, and data services. Recently, the quality of service (QoS) for real-time traffic handling in wireless LANs has become another important factor in wireless network medium access control research. A primary goal in this area is to

guarantee the required QoS for real-time traffic while providing high throughput for best-effort data traffic[8, 91]. Providing quality of service (QoS) is particularly challenging in networks that include wireless links. The quality of a wireless channel is typically different for different users, and randomly changes in time with both slow and fast time scales. In addition, wireless link capacity is usually a scarce resource that needs to be used efficiently. Therefore, it is important to find efficient ways of supporting QoS for real-time data over wireless channels. Efficient scheduling is one of the ways to address the issue described above. It considers the problem of scheduling transmissions of multiple real-time applications sharing the same wireless channel so as to satisfy the desired delay constraints or throughput constraints of all users.

QoS based MAC should provide an efficient use of the available bandwidth while satisfying the quality-of-service (QoS) requirements of both data and real-time applications. The capacity of the network and QoS depend critically on the performance of the MAC protocol in terms of packet dropping rate, delay, throughput, and utilization. A MAC protocol for supporting QoS distinguishes itself from other MAC protocols in that various mechanisms are required to handle the diverse traffic demands of different services such as constant bit rate (CBR), variable bit rate (VBR), and available bit rate (ABR). CBR traffic such as voice telephony, VBR traffic such as video conferencing, and ABR traffic such as file data have very different service requirements in terms of delay and loss tolerance and throughput. Multiplexing these diverse services with reasonable quality of service (QoS) while maximizing the utilization of the channel bandwidth is a challenging task. Thus, while traditional ALOHA type MAC protocols can handle homogeneous traffic efficiently, different techniques are needed for a wireless MAC system which supports QoS. The transmission mechanism affects significantly the performance of a MAC protocol. Carrier sense multiple access (CSMA) is one of

the most pervasive MAC schemes in wireless networks, however, it does not provide QoS guarantees for real-time traffic support. The IEEE 802.11 standard, which is based on CSMA, allows the support of both best-effort and continuous media type of services, but fails in efficiently providing quality of service (QoS) for different traffic classes in a multi cell scenario. The need for a suitable MAC protocol is therefore clear.

This dissertation is organized as follows. In the next Chapter, we propose a new collision resolution algorithm called fast collision resolution (FCR) from which we develop a new MAC protocol. Detailed performance evaluation is carried out. In Chapter 3, we present the fairly scheduled FCR to address the fairness issue. A new QoS based MAC protocol will be investigated in Chapter 4. We then present the conclusions and future research directions in the last Chapter.

CHAPTER 2

Fast Collision Resolution (FCR) Algorithms

In this chapter, distributed contention-based algorithms will be discussed according to the specific performance characteristics. In a distributed network, all data transmission and reception have to be in the same frequency band since there are no special stations to translate the transmission from one frequency band to another. The IEEE802.11 MAC[53] is the representative distributed contention-based medium access control protocol widely used in current wireless LANs. The recently proposed Dynamic Tuning Backoff[18] algorithm improves the throughput performance of IEEE802.11 MAC by dynamically assigning the proper contention window size to each station based on a run-time estimation of the number of active stations. In what follows, we describe the basic operating procedures for these medium access control algorithms to facilitate comparative study with the proposed fast collision resolution (FCR) algorithm.

2.1 Distributed Contention-Based MAC Algorithms

2.1.1 ALOHA and Slotted ALOHA

In a pure ALOHA system, stations are allowed access to the channel whenever they have data to transmit. Each station monitors its transmission and waits for an acknowledgment from the destination station. By the receipt of an acknowledgment or by the lack of an acknowledgement, the transmitting station can determine the success or failure of a packet transmission attempt. If the transmission was unsuccessful, the station retransmits after a random amount of time to avoid the future collisions. One of the problem inherent in ALOHA system is packet retransmission delay. When a transmitted packet collides with packets from other users,

all of these packets are discarded and are retransmitted after some time interval. This time interval is decided randomly to avoid further collision among retransmitted packets. Packet retransmission may be repeated causing huge amount of packet delay. This problem arises especially when the channel traffic increases and approaches the channel capacity.

Among the protocols for random access through wireless communication channel, slotted ALOHA is one of the most popular forms. In the slotted ALOHA system, there is a large population of identical users share a single communication channel, and the channel is divided into fixed size time slots. Packetized data, the length of which does not exceed the time slot duration, are transmitted only at the beginning of a slot. By means of this synchronous transmission, the slotted ALOHA system achieves channel capacity (0.368) twice as much as that of the pure ALOHA system.

2.1.2 IEEE 802.11 standard Medium Access Control

The IEEE 802.11 standard places specifications on the parameters of both the physical (PHY) and medium access control (MAC) layers of the network. The PHY layer, which actually handles the transmission of data between stations, can use either direct sequence spread spectrum, frequency-hopping spread spectrum, or infrared (IR) pulse position modulation. IEEE 802.11 makes provisions for data rates of 1 - 11 Mbps, and calls for operation in the 2.4 - 2.4835 GHz frequency band (in the case of spread-spectrum transmission), which is an unlicensed band for industrial, scientific, and medical (ISM) applications.

The MAC layer is a set of protocols which is responsible for maintaining order in the use of a shared medium. The 802.11 standard specifies a carrier sense multiple access with collision avoidance (CSMA/CA) protocol. In this protocol, when a station receives a packet to be transmitted, it first listens to ensure no other station is transmitting. If the channel is clear, it then transmits the packet.

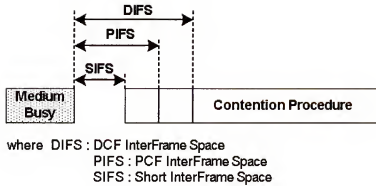


Figure 2.1: Inter Frame Spaces in IEEE 802.11 MAC

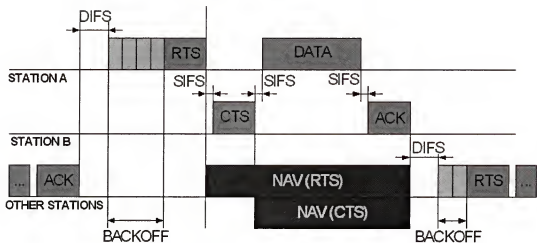


Figure 2.2: Basic operations of CSMA/CA

Otherwise, it chooses a random backoff time which determines the amount of time the station must wait until it is allowed to transmit its packet. In the optional RTS-CTS handshaking mechanism, the transmitting station first sends out a short ready-to-send (RTS) packet containing information on the length of the packet. If the receiving station hears the RTS, it responds with a short clear-to-send (CTS) packet. After this exchange, the transmitting station sends its packet. When the packet is received successfully, the receiving station transmits an acknowledgment (ACK) packet. The basic operations of the CSMA/CA algorithm are shown in Figure 2.1 and 2.2.

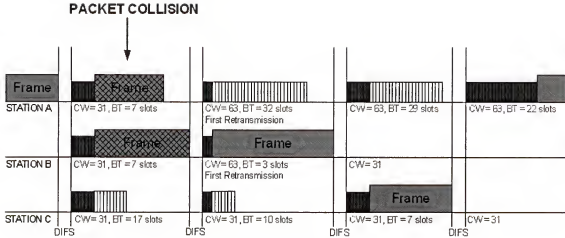


Figure 2.3: Random Backoff Procedure

A packet transmission cycle is accomplished with a successful transmission of a packet by a source station with an acknowledgment (ACK) from the destination station. More detailed operations of the IEEE 802.11 MAC protocol are as follows (we only consider distributed coordination function (DCF) without RTS-CTS handshake for simplicity). If a station has a packet to transmit, it will check the medium status by using the carrier sensing mechanism. If the medium is idle, the transmission may proceed. If the medium is determined to be busy, the station will defer until the medium is determined to be idle for a distributed coordination function inter-frame space (DIFS) and the backoff procedure will be invoked. The station will set its backoff timer to a random backoff time based on the current contention window size (CW):

$$\text{Backoff Time (BT)} = \text{Random}() \times \text{aSlotTime} \quad (2.1)$$

where $\text{Random}()$ is an integer randomly chosen from a uniform distribution over the interval $[0, \text{CW}-1]$.

After DIFS idle time, the station performs the backoff procedure by using the carrier sensing mechanism to determine whether there is any activity during each backoff slot. If the medium is determined to be idle during a particular backoff

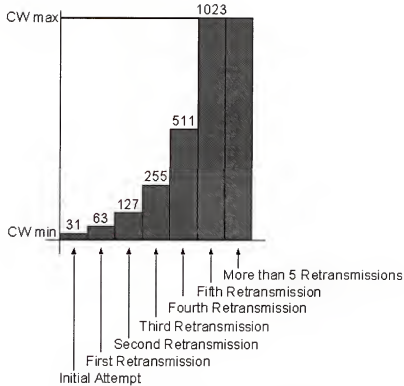


Figure 2.4: Binary Exponential Backoff

slot, then the backoff procedure shall decrement its backoff time by a slot time ($BT_{new} = BT_{old} - aSlotTime$). If the medium is determined to be busy at any time during a backoff slot, then the backoff procedure is suspended. After the medium is determined to be idle for DIFS period, the backoff procedure is allowed to resume. Transmission shall begin whenever the backoff timer reaches zero. After a source station transmits a packet to a destination station, if the source station receives an acknowledgment (ACK) without errors after short inter-frame space (SIFS) idle period, the transmission procedure is concluded to be successfully completed. If the transmission is successfully completed, the contention window (CW) for the source station shall be reset to the initial (minimum) value minCW. If the transmission is not successfully completed (i.e., the source station does not receive the ACK after SIFS), the contention window (CW) size shall be increased (in the IEEE 802.11 DSSS $CW = 2^{(n+5)} - 1$, *retry counter* $n = 0, \dots, 5$), beginning with the initial

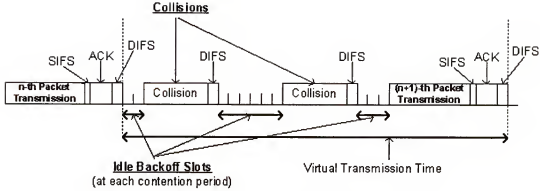


Figure 2.5: Access Scheme

value minCW , up to the maximum value maxCW (in the IEEE 802.11 DSSS, $\text{minCW}=31$ and $\text{maxCW}=1023$). This process is called *binary exponential backoff (BEB)*, which resolves collisions in the contention cycle. More detailed operations can be found in the reference[53].

2.1.3 Dynamic Tuning Backoff

Cali et al.[18] derive the average size of the contention window that maximizes the aggregate throughput, under the assumption that all stations have the same average contention window size of transmitting a packet in steady state. They assume that a station, in steady state, transmits a packet with the probability of $p = 1/(E[B] + 1)$, where $E[B]$ is the average value of the backoff timer. Since the average value of the backoff timer can be expressed as $E[B] = (E[CW] - 1)/2$, where $E[CW]$ is the average contention window size of sending a packet, the probability for a packet transmission is obtained by using the average contention window size as $p = 2/(E[CW] + 1)$.

The IEEE 802.11 protocol capacity is given as

$$\rho_{\max} = \frac{\bar{m}}{t_v} \quad (2.2)$$

where t_v is the average length of the renewal period, also referred to as the average virtual transmission time in Figure 2.5, and $\bar{m}$ is the average message length, i.e.,

the average time interval in a renewal period in which the channel is busy due to a successful transmission.

The throughput for one active station ρ_{single} is calculated by using $t_v = E[S] + E[B_1]$ where $E[S]$ is the time required to complete a successful transmission, $E[B_1]$ is the average backoff time.

$$\rho_{single} = \frac{\bar{m}}{2 \cdot \tau + \bar{m} + SIFS + ACK + DIFS + E[B_1]} \quad (2.3)$$

where $\bar{m}$ is the average transmission time and $E[B_1] = (CW_{min} - 1)/2$.

When multiple stations exist, the virtual transmission time includes a successful transmission and collision intervals. Therefore it follows

$$t_v = E\left\{\sum_{i=1}^{N_c} (Idle.p_i + Coll_i + \tau + DIFS)\right\} + E[Idle.p_{N_c+1}] + E[S] \quad (2.4)$$

where $Idle.p_i$ and $Coll_i$ are the lengths of the i th idle period and collision in a virtual transmission time, respectively; and N_c is the number of collisions in a virtual time.

Considering the idle period times are i.i.d. with an average $E[Idle.p]$, and the collision lengths are i.i.d with average $E[Coll]$, then, (2.4) can be rewritten as

$$t_v = E[N_c]\{E[Coll] + \tau + DIFS\} + E[Idle.p] \cdot (E[N_c] + 1) + E[S] \quad (2.5)$$

and

$$E[N_c] = \frac{1 - (1-p)^M}{Mp(1-p)^{M-1}} - 1 \quad (2.6)$$

$$E[coll] = \frac{t_s}{1 - [(1-p)^M + Mp(1-p)^{M-1}]} \cdot \left[\sum_{h=1}^{\infty} \{h \cdot [(1-pq^h)^M - (1-pq^{h-1})^M]\} - \frac{Mp(1-p)^{M-1}}{(1-q)} \right] \quad (2.7)$$

$$E[Idle.p] = \frac{(1-p)^M}{1 - (1-p)^M} \cdot t_{slot} \quad (2.8)$$

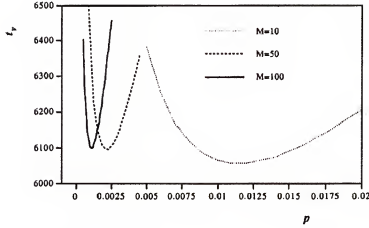


Figure 2.6: t_v function for different M values

Based on the iterative procedures, Cali et al. are able to derive the following formula for the aggregate network throughput ρ :

$$\rho = \frac{\bar{m}}{\frac{(1-p)}{Mp} t_s + \frac{1-(1-p)^M - Mp(1-p)^{M-1}}{Mp(1-p)^{M-1}} [E[coll] + \tau + DIFS] + E[S]} \quad (2.9)$$

where $\bar{m}$ is the average packet length, M is the number of active stations, τ is the maximum propagation time, q is the parameter for the geometric distribution of packet length, t_s is the length of a slot (i.e., aSlotTime), $E[coll]$ is the average collision length, and $E[S]$ is the average time to complete a successful packet transmission without any collisions.

Now, the aggregate network throughput ρ is derived as a function of the probability of a packet transmission p and the number of active stations M from (2.9), because all other parameters (τ , t_s , $\bar{m}$, q) are determined by the simulation configurations. This means that if the number of active stations M is fixed and given, then we can obtain the optimal p value which maximizes the network throughput, and this maximum throughput is the theoretical throughput limit or analytical upper bound based on the DTB analysis approach[18].

In the DTB algorithm, the throughput of the IEEE 802.11 MAC protocol, with an optimal backoff window size tuned to the optimal p value for each M , can be improved significantly. However, the p value, and hence the optimal contention window size of transmitting a packet, depends on the number of active stations. The DTB method needs to compute the optimal contention window size of transmitting a packet at run-time by estimating the number of active stations. If the estimation is not accurate, the wasting slots or packet collisions will be significant. However, to accurately estimate the number of active stations at run-time is not an easy task for practical wireless local area networks running a distributed contention-based MAC protocol. Moreover, we can achieve higher throughput results than the theoretical throughput limits which Cali et al. provided as analytical upper bounds based on their analytical model of contention-based MAC algorithms[18]. We will discuss a new approach to obtain the new maximum throughput for the contention-based MAC algorithms in the next chapter.

2.2 Fast Collision Resolution (FCR) MAC Algorithm

2.2.1 The Basic Idea

There are two major factors affecting the throughput performance in the IEEE 802.11 MAC protocol: transmission failures (we only consider failures due to packet collisions) and the idle slots due to backoff at each contention cycle, which are shown in Figure 2.5.

Under high traffic load (i.e., all M stations always have packets to transmit) and under some ergodicity assumption, we can obtain the following expression for the throughput (for example, based on Figure 2.5, we can examine one transmission cycle)[13, 18]:

$$\rho = \frac{\bar{m}}{E[N_c](E[B_c] \cdot t_s + \bar{m} + DIFS) + (E[B_c] \cdot t_s + \bar{m} + SIFS + ACK + DIFS)} \quad (2.10)$$

where $E[N_c]$ is the average number of collisions in a virtual transmission time (or a virtual transmission cycle), $E[B_c]$ is the average number of idle slots resulting from backoff for each contention period, t_s is the length of a slot (i.e., aSlotTime), and $\bar{m}$ is the average packet length.

From this result, we can see that the best scenario in Figure 2.5, which gives the maximum throughput, would be the following: a successful packet transmission must be followed by another packet transmission without any overheads, in which case, $E[N_c] = 0$, $E[B_c] = 0$, the throughput would be

$$\rho_{best} = \frac{\bar{m}}{(\bar{m} + SIFS + ACK + DIFS)} \quad (2.11)$$

This can be achieved only when a perfect scheduling is provided with an imaginable helping hand. In such a scenario, each station shall have the probability of packet transmission, $p_{trans}(i)$, at each contention period as follows:

$$p_{trans}(i) = \begin{cases} 1 & \text{if station } i \text{ transmits its packet at current contention period} \\ 0 & \text{otherwise} \end{cases} \quad (2.12)$$

Suppose that under some contention-based random backoff schemes, we could assume that the backoff timer is chosen randomly, then the probability of packet transmission for station i during the current contention period would depend on the backoff timer:

$$p_{trans}(i) = \frac{1}{(B_i + 1)} \quad (2.13)$$

where B_i is the backoff timer of station i .

This means that if station i has the backoff timer 0 (i.e., $B_i = 0$), then its backoff time is 0 and station i will transmit a packet immediately. Therefore, this can be interpreted as that station i has the probability of packet transmission of 1 at current contention period. If station i has the backoff timer ∞ , then its backoff time is also ∞ , which can be interpreted as that station i has the probability of

packet transmission of 0 at current contention period. From this discussion, (2.12) can be converted to (2.14):

$$B_i = \begin{cases} 0 & \text{if station } i \text{ transmits its packet at current contention period} \\ \infty & \text{otherwise} \end{cases} \quad (2.14)$$

Thus, we conclude that if we could develop a contention-based MAC algorithm, which assigns a backoff timer 0 to the station in transmission while assigns all other stations' backoff timers ∞ for each contention period, then we could achieve the perfect scheduling, leading to the maximum throughput. Unfortunately, such a contention-based MAC algorithm does not exist in practice. However, this does provide us the basic idea how to improve the throughput performance in the MAC protocol design. We can use the operational characteristics of the perfect scheduling to design more efficient contention-based MAC algorithm. One way to do so is to design an MAC protocol to approximate the behavior of perfect scheduling.

From (2.12) and (2.14), we conclude that to achieve high throughput, the MAC protocol should have the following operational characteristics:

1. *Small random backoff timer for the station which has successfully transmitted a packet at current contention cycle:* This will decrease the average number of idle slots for each contention period, $E[B_c]$ in (2.10).
2. *Large random backoff timer for stations that are deferred their packet transmissions at current contention period:* The deferred station means a station which has non-zero backoff timers. Large random backoff timers for deferred stations will decrease the collision probability at subsequent contention periods (and avoid future collisions more effectively).
3. *Fast change of random backoff timer according to its current state: transmitting or deferring:* When a station transmits a packet successfully, its random backoff timer should be set small. The net effect of this operation is

that whenever a station seizes the channel, it will use the medium as long as possible to increase the useful transmissions. When the station is deferred, its random backoff timers should be as large as possible to avoid the future collisions. The net effect is that all deferred stations will give the successful station more time to finish the back-logged packets. When a deferred station detects the medium is idle for a fixed number of slots, it would conclude that no other stations are transmitting, and hence it will reduce the backoff timers exponentially to reduce the average idle slots.

2.2.2 Fast Collision Resolution (FCR) Algorithm

As we pointed out, the major deficiency of the IEEE 802.11 MAC protocol comes from the slow collision resolution as the number of active stations increases. An active station can be in two modes at each contention period, namely, the transmitting mode when it wins a contention and the deferring mode when it loses a contention. When a station transmits a packet, the outcome is either one of the two cases: a successful packet transmission or a collision. Therefore, a station will be in one of the following three states at each contention period: a successful packet transmission state, a collision state, and a deferred state. In most distributed contention-based MAC algorithms, there is no change in the contention window size for the deferring stations, and the backoff timer will decrease by one slot whenever an idle slot is detected. In the proposed fast collision resolution (FCR) algorithm, we will change the contention window size for the deferring stations and regenerate the backoff timers for all potential transmitting stations to actively avoid “future” potential collisions, in this way, we can resolve possible packet collisions quickly. More importantly, the proposed algorithm preserves the simplicity for implementation like the IEEE 802.11 MAC.

The FCR algorithm has the following characteristics:

1. Use much smaller initial (minimum) contention window size $minCW$ than the IEEE 802.11 MAC;
2. Use much larger maximum contention window size $maxCW$ than the IEEE 802.11 MAC;
3. Increase the contention window size of a station when it is in both collision state and deferring state;
4. Reduce the backoff timers exponentially fast when a prefixed number of consecutive idle slots are detected.
5. Assign the maximum successive packet transmission limit to keep fairness in serving users.

Item 1 and 4 attempt to reduce the average number of idle backoff slots for each contention period ($E[B_c]$) in (2.10). Items 2 and 3 are used to quickly increase the backoff timers, hence quickly decrease the probability of collisions. In item 3, the FCR algorithm has the major difference from other contention-based MAC protocols such as the IEEE 802.11 MAC. In the IEEE 802.11 MAC, the contention window size of a station is increased only when it experiences a transmission failure (i.e., a collision). In the FCR algorithm, the contention window size of a station will increase not only when it experiences a collision but also when it is in the deferring mode and senses the start of a new busy period. Therefore, all stations which have packets to transmit (including those which are deferred due to backoff) will change their contention window sizes at each contention period in the FCR algorithm. Item 5 is used to avoid that a station dominates packet transmissions for a long period. If a station has performed successive packet transmissions of the maximum successive packet transmission limit, it changes its contention window size to the maximum value ($maxCW$) to give opportunities for medium access to other stations.

The detailed FCR algorithm is described as follows according to the state a station is in:

1. *Backoff Procedure*: All active stations will monitor the medium. If a station senses the medium idle for a slot, then it will decrement its backoff time (BT) by a slot time, i.e., $BT_{new} = BT_{old} - aSlotTime$ (or the backoff timer is decreased by one unit in terms of slot). When its backoff timer reaches to zero, the station will transmit a packet. If there are $[(minCW + 1) \times 2 - 1]$ consecutive idle slots being detected, its backoff timer should be decreased much faster (say, exponentially fast), i.e., $BT_{new} = BT_{old} - BT_{old}/2 = BT_{old}/2$ (if $BT_{new} < aSlotTime$, then $BT_{new} = 0$) or the backoff timer is decreased by a half. For example, if a station has the backoff timer 2047, hence its backoff time is $BT = 2047 \times aSlotTime$, which will be decreased by a slot time at each idle slot until the backoff timer reaches 2040 (we assume that $[(minCW + 1) \times 2 - 1] = 7$ or $minCW = 3$). After then, if the idle slots continue, the backoff timer will be decreased by one half, i.e., $BT_{new} = BT_{old}/2$ at each additional idle slot until either it reaches to zero or it senses a non-idle slot, whichever comes first. As an illustration, after 7 idle slots, we will have $BT = 1020 \times aSlotTime$ on the 8th idle slot, $BT = 510 \times aSlotTime$ on the 9th idle slot, $BT = 255 \times aSlotTime$ on the 10th idle slot, and so on until it either reaches to zero or detects a non-idle slot. Therefore, the wasted idle backoff time is guaranteed to be less than or equal to $18 \times aSlotTime$ for above scenario. The net effect is that the unnecessary wasted idle backoff time will be reduced when a station, which has just performed a successful packet transmission, runs out of packets for transmission or reaches its maximum successive packet transmission limit.
2. *Transmission Failure (Packet Collision)*: If a station notices that its packet transmission has failed possibly due to packet collision (i.e., it fails to receive

an acknowledgment from the intended receiving station), the contention window size of the station will be increased and a random backoff time (BT) will be chosen, i.e., $CW = \min(maxCW, CW \times 2)$, $BT = \text{uniform}(0, CW - 1) \times aSlotTime$, where $\text{uniform}(a, b)$ indicates a number randomly drawn from the uniform distribution between a and b and CW is the current contention window size.

3. *Successful Packet Transmission*: If a station has finished a successful packet transmission, then its contention window size will be reduced to the initial (minimum) contention window size $minCW$ and a random backoff time (BT) value will be chosen accordingly, i.e., $CW = minCW$, $BT = \text{uniform}(0, CW - 1) \times aSlotTime$. If a station has performed successive packet transmissions which reaches the maximum successive transmission limit (or larger), then its contention window size will be increased to the maximum contention window size $maxCW$ and a random backoff time (BT) value will be chosen as follows: $CW = maxCW$, $BT = \text{uniform}(0, CW - 1) \times aSlotTime$.
4. *Deferring State*: For a station which is in deferring state, whenever it detects the start of a new busy period, which indicates either a collision or a packet transmission in the medium, the station will increase its contention window size and pick a new random backoff time (BT) as follows: $CW = \min(maxCW, CW \times 2)$, $BT = \text{uniform}(0, CW - 1) \times aSlotTime$.

In the FCR algorithm, the station that has successfully transmitted a packet will have the minimum contention window size and smaller backoff timer, hence it will have a higher probability to gain access of the medium, while other stations have relatively larger contention window size and larger backoff timers. After a number of successful packet transmissions for one station, another station may win

Table 2.1: Example of IEEE 802.11 MAC with binary exponential backoff

0	1	2	3	4	5	6	7	8	9	Station Number
1(7)	3(7)	2(7)	7(7)	2(7)	6(7)	3(7)	4(7)	1(7)	6(7)	Contention Begins
~~0(7)~~	2(7)	1(7)	6(7)	1(7)	5(7)	2(7)	3(7)	~~0(7)~~	5(7)	Collision on station 0 & 8
8(15)								14(15)		
7(15)	1(7)	~~0(7)~~	5(7)	~~0(7)~~	4(7)	1(7)	2(7)	13(15)	4(7)	Collision on station 2 & 4
		4(15)		9(15)						
6(15)	~~0(7)~~	3(15)	4(7)	8(15)	3(7)	~~0(7)~~	1(7)	12(15)	3(7)	Collision on station 1 & 6
	10(15)					5(15)				
5(15)	9(15)	2(15)	3(7)	7(15)	2(7)	4(15)	~~0(7)~~	11(15)	2(7)	Successful Packet Transmission on station 7
							3(7)			

* Each item indicate: Backoff Timer B_i (Contention Window Size)

a contention and this new station will then have higher probability to gain access of the medium for a period of time.

To elaborate the operations of the FCR algorithm, we use some examples to illustrate the major difference between the IEEE 802.11 MAC and FCR algorithm. Table 2.1 shows an example of the IEEE 802.11 MAC operations with the contention window size $CW = 2^{(n+3)} - 1$, *retry counter* $n = 0, \dots, 7$ (i.e., $\min CW = 7$ and $\max CW = 1023$).

In this example, there are 10 active stations contending for the use of the medium based on the IEEE 802.11 MAC. When the contention begins (i.e., the medium is determined to be idle for DIFS period by the carrier sensing mechanism), each station performs the backoff procedure with its random backoff time (BT) determined from the initial contention window range $[0, 7]$ (hence $BT = \text{uniform}[0, 7] \times aSlotTime$). When a station detects the current slot idle, it will decrement its backoff time by a slot time $BT_{new} = BT_{old} - aSlotTime$ (i.e., the backoff timer is decreased by one unit). After one idle slot, the backoff timers of stations 0 and 8 reach to zero, thus in the following slot, both station 0 and station 8 will transmit their packets at the same time and a collision will occur. The backoff procedures of all deferred stations are suspended and will resume after the medium is determined to be idle for DIFS period (i.e., next contention

Table 2.2: Example of Fast Collision Resolution Algorithm

0	1	2	3	4	5	6	7	8	9	Station Number
1(3)	0(3)	2(3)	1(3)	2(3)	2(3)	3(3)	3(3)	1(3)	0(3)	Collision on 1 & 9
1(7)	3(7)	2(7)	7(7)	2(7)	6(7)	3(7)	4(7)	1(7)	6(7)	
0(7)	2(7)	1(7)	6(7)	1(7)	5(7)	2(7)	3(7)	0(7)	5(7)	Collision on 0 & 8
8(15)	10(15)	2(15)	1(15)	12(15)	4(15)	15(15)	6(15)	14(15)	3(15)	
7(15)	9(15)	1(15)	0(15)	11(15)	3(15)	14(15)	5(15)	13(15)	2(15)	Success on 3
22(31)	18(31)	28(31)	1(3)	5(31)	17(31)	11(31)	9(31)	14(31)	23(31)	
21(31)	17(31)	27(31)	0(3)	4(31)	16(31)	10(31)	8(31)	13(31)	22(31)	Success on 3
40(63)	9(63)	38(63)	3(3)	58(63)	24(63)	17(63)	20(63)	44(63)	1(63)	
39(63)	8(64)	37(63)	2(3)	57(63)	23(63)	16(63)	19(63)	43(63)	0(63)	Success on 9
100(127)	55(127)	29(127)	5(7)	111(127)	46(127)	81(127)	30(127)	9(127)	1(3)	
99(127)	54(127)	28(127)	4(7)	110(127)	45(127)	80(127)	29(127)	8(127)	0(3)	Success on 9
67(255)	29(255)	189(255)	11(15)	55(255)	210(255)	160(255)	240(255)	120(255)	2(3)	

* Each item indicate: Backoff Timer B_i (Contention Window Size)

period). After stations 0 and 8 notice that their packet transmissions fail, their contention window sizes will be increased to 15 and their backoff timers will be chosen in the range of $[0, 15]$ randomly. When a new DIFS period is detected, stations 2 and 4 transmit packets after one idle slot and a collision occurs. Stations 1 and 6 transmit packets and a collision occurs in the following contention period. After then, when the next DIFS period is detected, station 7 has a successful packet transmission. In the whole contention cycle (the time period starting with the end of a successful packet transmission and ending with the start of the next successful packet transmission), there have been three consecutive collisions before one successful packet transmission. We observe in Table 2.1 that most contention window sizes chosen for the backoffs are not big enough to avoid future packet collisions. Since the IEEE 802.11 MAC cannot provide the proper contention window size as the number of active stations increases, collisions are not resolved quickly, which leads to poor throughput performance.

Table 2.2 shows an example for the FCR algorithm with the contention window size $CW = 2^{(n+2)} - 1$, *retry counter* $n = 0, \dots, 9$ (i.e., $\min CW = 3$ and $\max CW = 2047$).

In Table 2.2, stations 1 and 9 collide in the first contention period. Stations 1 and 9 then increase their contention window sizes to 7 and pick up their backoff

timers in the range of $[0, 7]$ randomly. All deferring stations also increase their contention window sizes to 7 and pick up the new backoff timers in the range of $[0, 7]$ randomly. In the second contention period, stations 0 and 8 collide and will repeat the same procedure. In the third contention period, station 3 transmits a packet successfully. We observe in Table 2.2 that most contention window sizes of the deferring stations are increased quickly, so the FCR algorithm resolves the contentions very quickly, which results in significantly lower collision probability during each contention period in the future.

In Table 2.1 and Table 2.2, we can clearly see the major differences in operations between the IEEE 802.11 MAC and the FCR algorithm. To put it briefly, the high throughput of the FCR algorithm comes from: the small backoff time for the station that transmits a packet at current contention period (this reduces the wasted idle slots), the large backoff time for the stations which are deferred for packet transmissions (this reduces the collision probability), and faster change of backoff timers according to the current state: transmitting or deferred. This means that the FCR algorithm satisfies well the required condition for high throughput performance which is shown in (2.14).

2.2.3 Performance Analysis of FCR

In the asymptotic condition with high traffic load (i.e., stations always have packets to transmit), a station can have two situations, namely, transmitting situation when it wins in the contention procedure against other stations, and deferring situation during the time in which other stations win in the contention procedure and are sending packets. If we change the contention window size for the deferring situations, then we can achieve higher throughput than IEEE802.11 MAC algorithms. In this section, we will analyze the high throughput performance of FCR algorithm against the IEEE802.11 BEB and Dynamic Tuning Backoff

algorithms.

$$p = \frac{2}{E[CW]_{PkSend} + 1} \quad (2.15)$$

In DTB analysis, the probability of a packet transmission is derived by (2.15).

The procedure to find the average contention window size of sending a packet is as follows. By using the probability of a packet transmission p for one station, the probability of a successful packet transmission of the network at each stage is derived as $p_{suc} = (1 - p)^{M-1}$, where M is the number of active stations. Thus, a station has a successful packet transmission with the probability $p_{suc} = (1 - p)^{M-1}$ and a collision with the probability $p_{coll} = 1 - (1 - p)^{M-1}$. In the backoff algorithm for IEEE802.11 Frequency Hopping Spread Spectrum(FHSS) system, there are four different contention window size, 32, 64, 128, 256[53]. The probability of having a contention window size CW_j is

$$P[CW = CW_j] = \begin{cases} (1 - p_{coll})(p_{coll})^j & j = 0, 1, 2 \\ (p_{coll})^j & j = 3 \end{cases}$$

where $CW_0 = 32$, $CW_1 = 64$, $CW_2 = 128$, $CW_3 = 256$

Therefore, the expected contention window size of sending a packet is

$$E[CW]_{PkSend} = 256(p_{coll})^3 + (128(p_{coll})^2 + 64(p_{coll}) + 32)(1 - p_{coll}) \quad (2.16)$$

where $p_{coll} = 1 - (1 - p)^{M-1}$

If we use (2.15) and (2.16) for a given number of active stations M , we can estimate the expected contention window size of sending a packet $E[CW]_{PkSend}$ by using an iterative method. The aggregate network throughput is derived by using the probability of a packet transmission p and the number of active stations M , which is shown (2.9). The maximum throughput and the corresponding p for a given M value are obtained from (2.9) and it is claimed that this maximum

throughput is the analytical bound regardless of the particular type of backoff scheme in IEEE802.11 BEB based backoff algorithms.

In what follows, we carry out performance analysis for FCR algorithm which achieves higher throughput than DTB algorithm. To analyze the throughput capacity more precisely for FCR algorithm, we have to consider all possible states for the contention procedure, including a successful packet transmission, collisions, and deferred conditions. The information for the probability distribution of contention window sizes for sending a packet is needed to estimate the average contention window size of sending a packet when we consider the deferred conditions. We consider two kinds of contention window sizes, one for the whole contention procedure including deferring conditions and the other for the case of transmitting a packet. The relation between the average contention window size for each contention procedure $E[CW]$ and the average probability of successful packet transmission for one station is given by the following equation.

$$p_{suc,1} = \sum_{i=1}^{E[CW]} \frac{1}{E[CW]} \cdot \left(\frac{E[CW] - i}{E[CW]} \right)^{M-1} \quad (2.17)$$

Furthermore, the summation of the probability of collision and the probability of deferring for one station is given by the following equation. $1 - p_{suc,1} = p_{col,1} + p_{defer,1}$

In FCR, the contention window size for each contention procedure is increased by the increasing factor (IF) when a station experiences a collision or a deferred situation, and goes to the minimum value with a successful packet transmission. Therefore, the average contention window size for each contention procedure is

$$E[CW]^{i+1} = p_{suc,1} \times minCW + (1 - p_{suc,1}) \times E[CW]^i \times IF \quad (2.18)$$

If we use the above equations, we can use an iterative process to obtain the average contention window size for each contention procedure $E[CW]$ and

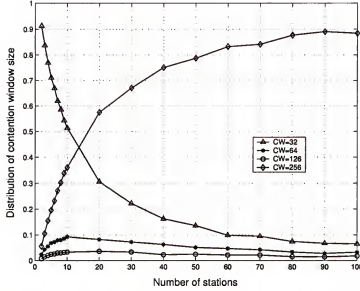


Figure 2.7: Distribution for contention window size of sending a packet

the probability of a successful packet transmission for one station $p_{suc,1}$. If the number of stations in the network is M , the total probability of successful packet transmission for the whole network is $p_{suc,total} = p_{suc,1} \cdot M$ and the total average probability of collision for the whole network is $p_{col,total} = 1 - p_{suc,total}$.

Now, we can calculate the average number of collisions for a successful packet transmission

$$E[N_c] = \frac{p_{col,total}}{p_{suc,total}} \quad (2.19)$$

To calculate the average idle backoff slot number $E[IdleSlot]/t_{slot}$, we need the probability of sending a packet at each contention window size $E[CW]_{PkSend}$. In Figure 2.7, the distribution of contention window size for sending a packet in steady state is shown for the $minCW = 32$, $maxCW = 256$, $IF = 2$ case. In the 10 station case, 51% of stations have the contention window size of sending a packet at $CW = 32$, 9% of stations have $CW = 64$, 3% of stations have $CW = 128$, and 37% of stations have $CW = 256$.

The average contention window size of sending a packet is

$$E[CW]_{PkSend} = 32 \times p_{PkSend,32} + 64 \times p_{PkSend,64} + 128 \times p_{PkSend,128} + 256 \times p_{PkSend,256} \quad (2.20)$$

The average number of idle backoff slots is given by the following equation

$$E[IdleSlot]/t_{slot} = \sum_{i=1}^{E[CW]_{PkSend}-1} i \times \frac{(E[CW]_{PkSend} - i)^{M-1}}{(E[CW]_{PkSend})^M} \quad (2.21)$$

$$\rho = \frac{\bar{m}}{(E[N_c] + 1) \times (E[IdleSlot] + \bar{m} + SIFS + ACK + DIFS)} \quad (2.22)$$

Finally, the throughput of FCR is given by (2.22). In this analytical approach, we consider the deferred state with a successful packet transmission and a collision. The main difference of FCR algorithm from other distributed contention-based MAC algorithms is the action taken on the deferred situation and this action leads to the attendant performance improvements.

2.2.4 Performance Results for IEEE 802.11 FHSS: 2 Mbps

We assume that the best-effort data packets are always available at all stations. In the simulations, the packet lengths for the best-effort data packets are geometrically distributed with parameter q [18]:

$$P[PacketLength = i \text{ slots}] = q^{i-1}(1 - q), \quad i \geq 1.$$

Thus, the average transmission time for a packet (the average packet length) is given by:

$$\bar{m} = t_s / (1 - q) \quad (\mu sec)$$

where t_s is the slot time, i.e., $t_s = aSlotTime$.

We assigned the maximum successive packet transmission limit of the FCR algorithm as 10. All simulations are performed for 100 second simulation time.

In this section, we present the simulation studies for the proposed fast collision resolution (FCR) algorithm in the frequency hopping spread spectrum(FHSS)

Table 2.3: Network Configurations: IEEE 802.11 FHSS 2Mbps

Parameter	Value
SIFS	28 μ sec
DIFS	128 μ sec
aSlotTime	50 μ sec
aPreambleLength	96 μ sec
aPLCPHeaderLength	32 μ sec
Bit rate	2 Mbps

wireless LANs[53]. The parameters used in the simulations are shown in Table 2.3.

In the simulations, we use the same simulation environments to compare the simulation results of the FCR algorithm with those simulation results from the DTB algorithm paper[18].

Table 2.4: Throughput Results for FCR Algorithm

(MinCW,MaxCW)	(3,2047)	(3,4095)	(3,1023)	(3,511)	(7,2047)	(15,2047)	(7,1023)
10 Data Station Case	0.7852	0.7795	0.7872	0.7833	0.7577	0.7033	0.7569
100 Data Station Case	0.7656	0.7792	0.7221	0.6507	0.7454	0.6662	0.7128

Table 2.5: Throughput Results for IEEE 802.11 MAC Algorithm

(MinCW,MaxCW)	(31,255)	(15,1023)
10 Data Station Case	0.6564	0.6075
100 Data Station Case	0.3197	0.3775

In Table 2.4 and 2.5, the throughput results of the FCR and IEEE 802.11 MAC algorithms with the average packet length of 40 slots are shown for various (MinCW, MaxCW) combinations. In Table 2.4, we can see that if we use large minimum contention window size (MinCW) of 7 and 15 in the FCR algorithm, then the throughput is decreased because of the wasting idle slots. If we use small maximum contention window size (MaxCW) of 1023 and 511, then the throughput is decreased because of the high collision probability under large number of users. Too large value of the MaxCW like 4095 also decreases the throughput of the FCR algorithm for small number of stations. From the simulation results, we

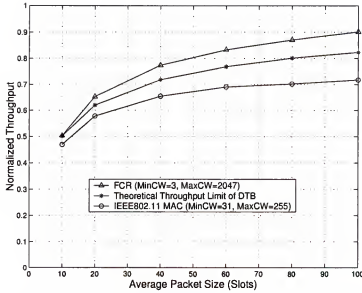


Figure 2.8: Throughput for 10 BE data stations wireless LAN

choose $\text{MinCW}=3$ and $\text{MaxCW}=2047$, which make a good throughput performance throughout different situations, as the basic parameters for the simulations of the FCR algorithm hereafter. The throughput results of the IEEE 802.11 MAC algorithm with two different (MinCW , MaxCW) combinations are shown in Table 2.5. Current IEEE 802.11 FHSS standard provides the minimum contention window size and the maximum contention window size as (15, 1023)[53], while (31, 255) is used in the DTB paper[18]. If we use (31, 255), the throughput is better for 10 data station case than the throughput result of using (15, 1023). For 100 data station case, the throughput of using (15, 1023) shows better result. In this simulation, we use $\text{MinCW}=31$ and $\text{MaxCW}=255$ as the basic parameters for the simulations of the IEEE 802.11 MAC algorithm to keep the same simulation environments[18]. All other parameters for the simulations are the same.

Figures 2.8, 2.9 and 2.10 show the throughput results of the IEEE 802.11 MAC, DTB, and FCR algorithms for 10, 50, and 100 contending stations, where the average packet length changes from 10 slots ($q = 0.9$) to 100 slots ($q = 0.99$). The IEEE 802.11 MAC algorithm shows very poor throughput performance as

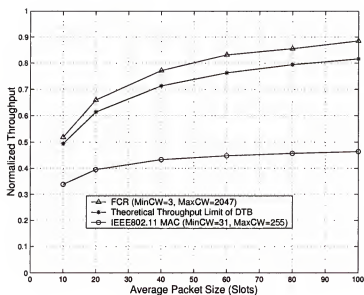


Figure 2.9: Throughput for 50 BE data stations wireless LAN

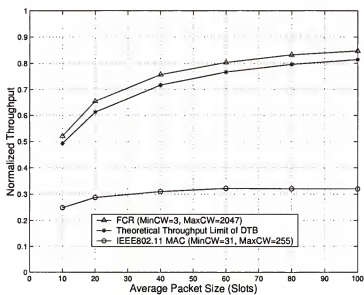


Figure 2.10: Throughput for 100 BE data stations wireless LAN

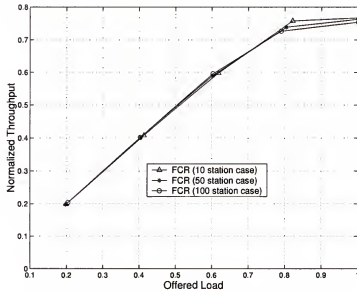


Figure 2.11: Throughput vs. offered load

the number of stations increases. The main reason is that the probability of collisions becomes higher as the number of stations becomes larger. In the FCR algorithm, all stations, except the one with successful packet transmission, will increase their contention window size whenever the system has either a successful packet transmission or has a collision. This means all stations can quickly obtain the proper contention window size to prevent future collisions, consequently the probability of collisions will be decreased to quite small values. At the same time, a station with a successful packet transmission has the minimum contention window size of 3, which is much smaller than the minimum contention window size in the IEEE 802.11 MAC algorithm ($\text{minCW}=31$). This will reduce the wasted medium idle time to a much smaller value when compared to the IEEE 802.11 MAC and the Dynamic Tuning Backoff algorithm. In Figures 2.8, 2.9 and 2.10, we can see that the FCR algorithm significantly improve the throughput performance over the IEEE 802.11 MAC algorithm. The FCR algorithm shows higher throughput performance than the theoretical throughput limit (the analytical upper bound) of the DTB algorithm. The FCR algorithm has much smaller wasting idle slots for

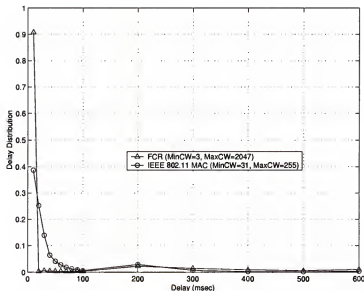


Figure 2.12: Delay distribution for 10 stations wireless LAN

each contention period than the DTB algorithm while both algorithms have similar values of the probability of collisions. Moreover, the throughput performance of the FCR algorithm are not severely degraded as the number of stations increases because of the highly efficient collision resolution strategy. Figure 2.11 shows the throughput vs. offered load for the FCR algorithm for 10, 50, 100 stations wireless LAN with the average packet length of 40 slots. We use a traffic generator with Poisson distribution to provide each offered load in this simulation. From Figure 2.11, we can see that the FCR algorithm also performs very efficiently under light load conditions while providing high throughput as network load increases, and the number of stations hardly affects the performance of the FCR algorithm.

We carry out the analysis for the packet delay of the IEEE 802.11 MAC and the FCR algorithm with the average packet length of 40 slots. The packet delay means the time period from the time when a packet arrives from higher layer to the MAC layer to the time it is successfully transmitted to the intended receiving station. Figures 2.12 and 2.13 show the packet delay distributions for the IEEE 802.11 MAC and the FCR algorithm for 10 and 100 stations wireless LANs. We

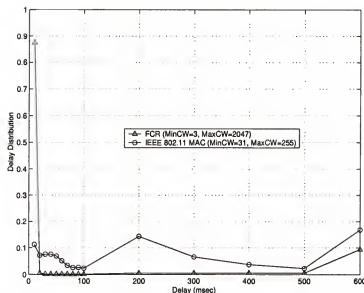


Figure 2.13: Delay distribution for 100 stations wireless LAN

have not apply limitation on the number of retries in this simulation for simplicity. In Figure 2.12, the FCR algorithm transmits 91% of all packets successfully within 10 msec while the remaining 9% packets spread over 10 msec to over 600 msec in delay distribution. However, the IEEE 802.11 MAC transmits 39% packets within 10 msec, 25% packets in the range from 10 msec to 20 msec, 13% packets in the range from 20msec to 30 msec, and so on. In Figure 2.13, the FCR algorithm transmits 88% of all packets successfully within 10 msec, while the IEEE 802.11 MAC transmits only 11% packets within 10 msec, 8% packets in the range from 10 msec to 20 msec, 8.5% packets in the range from 20 msec to 30 msec, and so on. In the simulation results for the packet delay, it is clear that the FCR algorithm transmits most packets successfully within pretty short time, while the IEEE 802.11 MAC transmits packets in much longer time due to collisions, which indeed shows that the FCR algorithm does resolve collision much more efficiently than the IEEE 802.11 MAC algorithm does.

Table 2.6: Network Configurations: IEEE 802.11 DSSS 2Mbps

Parameter	Value
SIFS	10 μ sec
DIFS	50 μ sec
A slot time	20 μ sec
aPreambleLength	144 bits
aPLCPHeaderLength	48 bits
Bit rate	2 Mbps

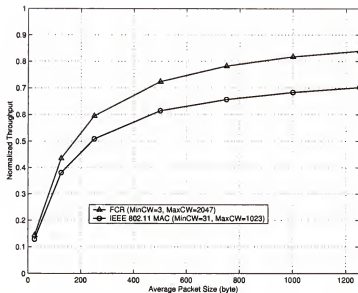


Figure 2.14: Throughput for 10 BE data stations wireless LAN

2.2.5 Performance Results for IEEE 802.11 DSSS: 2 Mbps

In this section, we present the simulation studies for the proposed fast collision resolution (FCR) algorithm and the IEEE 802.11 MAC protocol in a wireless LAN using direct sequence spread spectrum (DSSS). The parameters used in the simulations are shown in Table 2.6, which are based on the IEEE 802.11 network configurations[53].

Figures 2.14, 2.15 and 2.16 show the throughput results of the IEEE 802.11 MAC and FCR for 10, 50, and 100 contending stations, where the average transmission time for a packet (i.e., the average packet length) changes from 100 μ sec (25 bytes) to 5000 μ sec (1250 bytes).

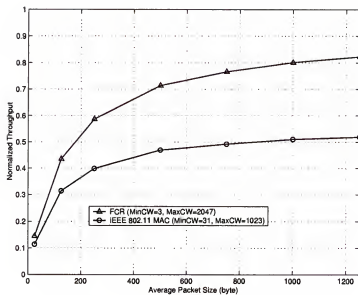


Figure 2.15: Throughput for 50 BE data stations wireless LAN

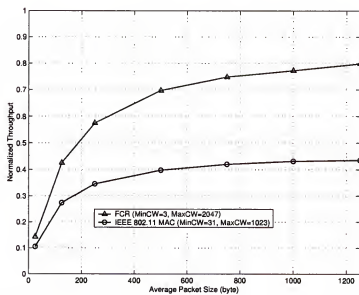


Figure 2.16: Throughput for 100 BE data stations wireless LAN

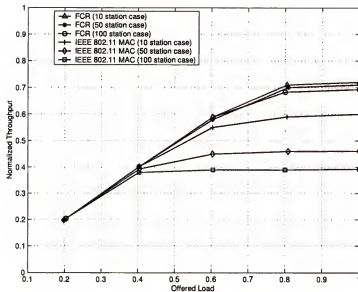


Figure 2.17: Throughput vs. offered load

Figure 2.17 shows the throughput vs. offered load for the IEEE 802.11 MAC and the FCR algorithm for 10, 50, 100 stations wireless LAN with the average transmission time for a packet (i.e., the average packet length) of $2000 \mu\text{sec}$ (500 bytes).

Figures 2.18 and 2.19 show the packet delay distributions for the IEEE 802.11 MAC and the FCR algorithm for 10 and 100 stations DSSS wireless LANs. In Figure 2.18, the FCR algorithm transmits 92% of all packets successfully within 10 msec while the remaining 8% packets spread over 10 msec to over 600 msec in delay. In Figure 2.19, the FCR algorithm transmits 89% of all packets successfully within 10 msec while the remaining 11% packets spread over 10 msec to over 600 msec in delay.

2.2.6 Performance Results for IEEE 802.11b DSSS: 11 Mbps

In this section, we present the simulation studies for the proposed fast collision resolution (FCR) algorithm and the IEEE 802.11b MAC protocol in a wireless LAN using direct sequence spread spectrum (DSSS). The parameters used in the simulations are shown in Table 2.7, which are based on the IEEE 802.11b network

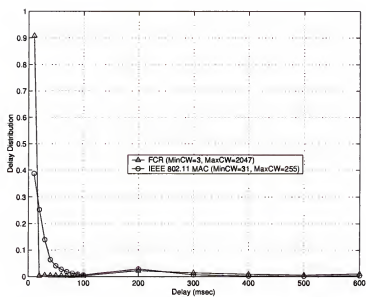


Figure 2.18: Delay distribution for 10 stations wireless LAN

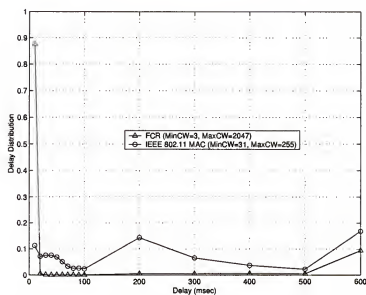


Figure 2.19: Delay distribution for 100 stations wireless LAN

Table 2.7: Network Configurations: IEEE 802.11 DSSS 2, 11Mbps

Parameter	Value
SIFS	10 μ sec
DIFS	50 μ sec
A slot time	20 μ sec
aPreambleLength	144 bits
aPLCPHeaderLength	48 bits
Bit rate	2, 11 Mbps

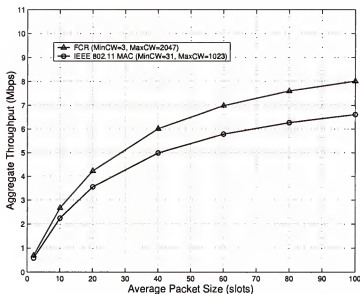


Figure 2.20: Throughput for 10 BE data stations wireless LAN

configurations. The transmission rates for data and ACK frame are 11 Mbps and 2 Mbps each.

Figures 2.20, 2.21 and 2.22 show the throughput results of the IEEE 802.11 MAC and FCR algorithms for 10, 50, and 100 contending stations, where the average transmission time for a packet (i.e., the average packet length) changes from 10 slots ($q = 0.9$) to 100 slots ($q = 0.99$).

Figure 2.23 shows the throughput vs. offered load for the IEEE 802.11 MAC and the FCR algorithm for 10, 50, 100 stations wireless LAN with the average transmission time for a packet (i.e., the average packet length) of 40 slots ($q = 0.975$).

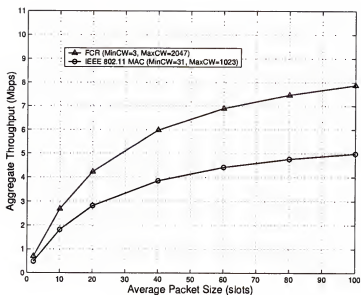


Figure 2.21: Throughput for 50 BE data stations wireless LAN

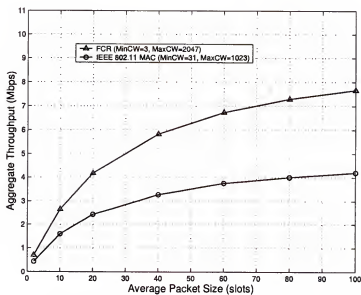


Figure 2.22: Throughput for 100 BE data stations wireless LAN

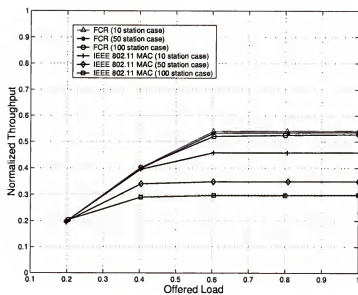


Figure 2.23: Throughput vs. offered load

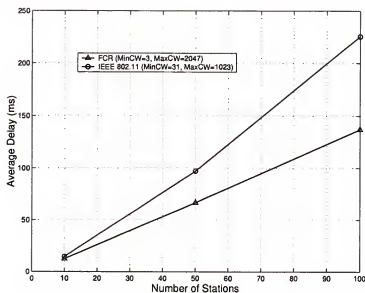


Figure 2.24: Average Delay

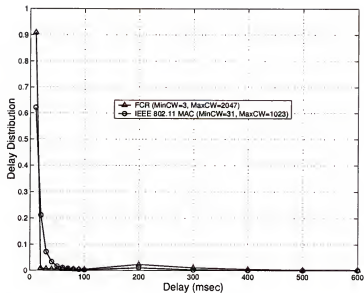


Figure 2.25: Delay distribution for 10 stations wireless LAN

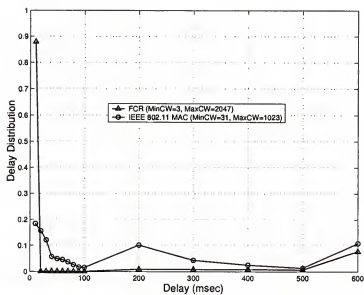


Figure 2.26: Delay distribution for 100 stations wireless LAN

Figures 2.24 shows the average delay of the IEEE 802.11 MAC and the FCR algorithm for 10, 50, and 100 stations wireless LANs. Figures 2.25 and 2.26 show the packet delay distributions for 10 and 100 stations. In Figure 2.25, the FCR algorithm transmits 91% of all packets successfully within 10 msec while the remaining 9% packets spread over 10 msec to over 600 msec in delay. However, the IEEE 802.11 MAC transmits 62% packets within 10 msec, 21% packets in the range from 10 msec to 20 msec, 7% packets in the range from 20msec to 30 msec, and so on. In Figure 2.26, the FCR algorithm transmits 88% of all packets successfully within 10 msec, while the IEEE 802.11 MAC transmits only 18% packets within 10 msec, 16% packets in the range from 10 msec to 20 msec, 12% packets in the range from 20 msec to 30 msec, and so on.

2.3 Effects on Performance at Transport Layer

2.3.1 Overview of Transport Layer Protocols

Transmission control protocol (TCP) and user datagram protocol (UDP) are the prevalent transport layer protocols which are used with the internet protocol (IP) of the network layer. They support transparent data transfer and perform flow and congestion control, ordering of received data, acknowledgment of correctly received data, etc. TCP and UDP run above the network and MAC layers, therefore, MAC layer protocols for wireless LANs should support TCP and UDP well. However, the low bandwidth and high error rate (even moderate packet loss rate) of the wireless channel can cause severe effects on the performance of the transport layer[103, 104]. The overheads of MAC layer may cause many retransmission segments which are not acknowledged within the retransmission time out (RTO) interval in the TCP operation, and result in performance degradation. Therefore, the evaluation of the proposed MAC algorithms for wireless LANs should be performed over transport layer as well as MAC layer.

The transport layer provides end-to-end communication services between different hosts. It makes available transparent data transfer using the services of the network layer below. Therefore, it generally supports various methods of flow control, error recovery and ordering of received data, acknowledgement of correctly received data, and multiplexing and demultiplexing sessions together. Applications and end users of the TCP/IP suite employ one of two protocols from transport layer: the transmission control protocol (TCP) or the user datagram protocol (UDP)[93]. We briefly explain the basic functions for these two protocols.

2.3.2 Transmission Control Protocol (TCP)

TCP is a pervasive transport protocol which gives a dependable data transfer service. It provides reliability for each end host by performing a connection oriented data transfer with supporting diverse flow and congestion controls as well as error recovery. If the data segments and acknowledgments are lost, that is, the sender can not receive an acknowledgment for a data segment within predetermined timeout interval, it retransmits the data segment. Therefore, the design strategy for timeout and retransmission has been the main issue to improve the TCP performance[93].

When delivering large amount of data, a sender should decide the transfer speed considering the receiver's buffer status to avoid network congestions and resulting data loss. Slow start is the procedure that can control the amount of data in-transit between sender and receiver. It works by monitoring the rate that new packets are transferred into the network and the rate that acknowledgments from the receiver are returned. The slow start mechanism counts on the sliding window and congestion window operations. The sliding window mechanism allows the sender to transmit multiple packets before it stops and waits for an acknowledgment. If a connection is established, the sending host transmits data to the receiving host. The receiver acknowledged with advertising its receiving

window size which allows the amount of data the sender can transmit. After the acknowledgement is received, the sender can send additional segments which is limited by the advertised window size of the receiver. Besides, a TCP sender manages its data transfer rate by using the congestion window(cwnd). When a new TCP connection is established, cwnd is set to one segment. Each time an ACK is received, the congestion window is increased by one segment. This phase is known as slow start. Therefore, the sender can transmit up to the minimum of the congestion window and the advertised window from the receiver.

If packets get lost because of packet damages in transit or network congestions, TCP operates flow control or congestion control algorithms. Congestion avoidance algorithm is a way to take care of lost packets. Congestion avoidance algorithm operates with slow start by maintaining the congestion window size and the slow start threshold size. If a segment is not acknowledged within some retransmission time out (RTO) interval, TCP performs retransmission to assure a reliable data delivery. If congestion occurs and RTO is expired, TCP assumes that a segment has been lost and retransmits it with setting the congestion window size as one segment and a slow start threshold as one-half of current window (but at least two segments). When new data is acknowledged by the receiver, either slow start or congestion avoidance is performed. If the congestion window is less than or equal to the slow start threshold, the slow start is triggered and increase the congestion window exponentially. Otherwise, congestion avoidance is triggered and the congestion window (cwnd) is increased by $1/cwnd$. Slow start continues until the congestion window arrives at the slow start threshold size. This is known as the congestion avoidance phase. If the same segment is lost consecutively, a backoff procedure is invoked, and the RTO is doubled after each retransmission.

If packet loss is detected, TCP slow start and congestion avoidance are performed and degrade the data throughput severely. To overcome this performance

degradation, fast retransmit and fast recovery have been designed to speed up the recovery of the connection. Fast retransmit and fast recovery detect a segment loss by monitoring duplicate acknowledgements. When a segment is lost, TCP at the receiver will keep sending ACK segments indicating the next expected sequence number which corresponds to the lost segment. The reception of three or more duplicate ACKs is a strong indication that a segment has been lost. Then, TCP fast retransmit mechanism carries out retransmission of the missing segment even before the retransmission timer expires. If only one or two packets are lost and there is still normal data flows between the two hosts, it is not necessary to reduce the transmission rate rapidly by using slow start. Therefore, fast recovery mechanism performs congestion avoidance instead of slow start, after a fast retransmit of the missing segment.

2.3.3 User datagram protocol (UDP)

User datagram protocol (UDP) is defined as a datagram mode of packet-switched computer communication and is a simple, datagram-oriented, connectionless, transport layer protocol[93]. UDP protocol supposes that the internet protocol (IP) is used in the network layer protocol, and performs a procedure for application programs to send messages with a minimum overhead of the protocol mechanism. UDP is transaction oriented, and delivery and duplicate protection are not assured. That is, it sends out the datagrams, but there is no guarantee that they ever reach the receiver. However, a lot of applications are better supported by using UDP because of no connection establishment, small packet overhead, and unfettered transmission rate. UDP encapsulates raw IP datagrams and sends them without having to establishing a connection. Many client-server applications that have one request and one response are much better suited for UDP rather than TCP which establishes and later releases a connection. Under UDP environments, the application is communicating almost directly with IP. UDP takes data packets

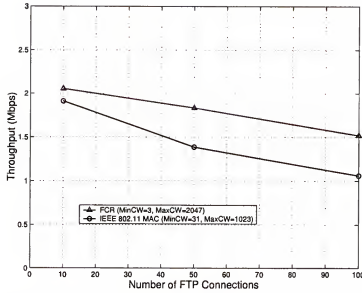


Figure 2.27: Throughput result for FTP traffic sources

from application process, attaches source and destination port number fields for the multiplexing/demultiplexing service, and passes the resultant segment to the network layer. The network layer encapsulates the segment into an IP datagram and then transfers the segment to the receiving host. If the segment comes to the receiving host, UDP deliver the data in the segment to the corresponding application process[93].

2.3.4 Performance Analysis

We run simulations to verify the efficiency of co-operations for the FCR algorithm with the transport layer protocols such as TCP and UDP by using the GlomoSim network simulator[9]. We checked the performance results on the transport layer by using different MAC layer protocols: IEEE 802.11 and FCR. In Figure 2.27 and 2.28, the throughput and fairness index for FTP connections are shown. The throughput, fairness and packet delivery ratio for constant bit rate (CBR) traffic are shown in Figure 2.29, 2.30, 2.31, 2.32 and 2.33. In Figure 2.27, the throughput result is shown for various number of FTP connections(10, 50, and 100). All FTP connections continuously send, from source stations to destination

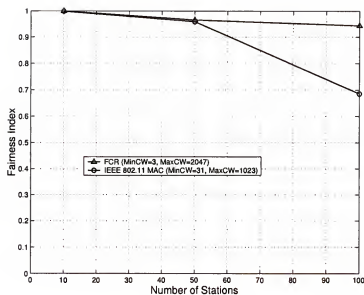


Figure 2.28: Fairness Index for FTP traffic sources

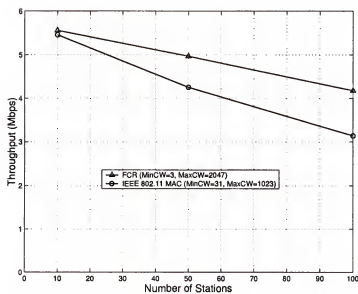


Figure 2.29: Throughput result for bursty CBR traffic sources

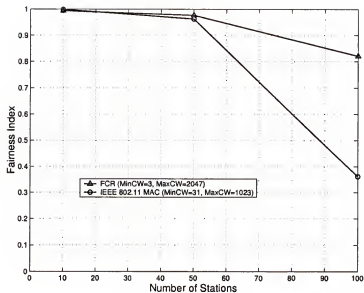


Figure 2.30: Fairness Index for bursty CBR traffic sources

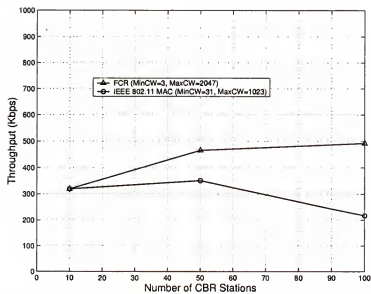


Figure 2.31: Throughput result for voice traffic sources

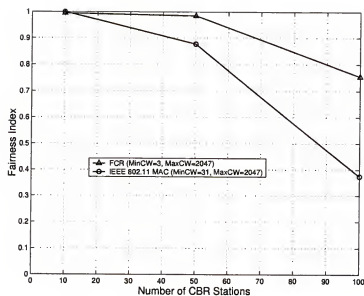


Figure 2.32: Fairness Index for voice traffic sources

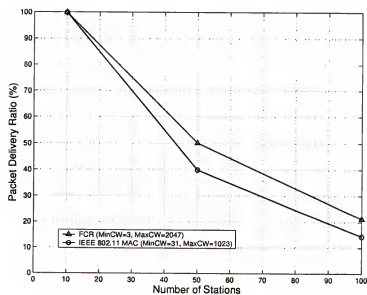


Figure 2.33: Packet Delivery Ratio

stations, data packets with 1460 bytes, and the simulation time is 100 sec. Figure 2.29 and 2.31 show the throughput results for CBR stations with the UDP operation. In Figure 2.29, CBR stations generate 1460 byte packets at every 1 *ms*. In Figure 2.31 and Figure 2.33, the throughput and the packet delivery ratio are shown for voice traffic stations (32 *kbps* CBR traffic). From the simulation results, we observe that the FCR algorithm improves the performance (fairness, aggregate throughput, packet delivery ratio) of the transport layer compared to the IEEE 802.11 MAC. This means the proposed FCR algorithm supports well the transport layer protocols such as TCP and UDP. Based on these simulation results, we can say that the efficient collision resolution scheme of the FCR algorithm can also significantly improves the performance at higher layers.

CHAPTER 3

FAIRLY SCHEDULED Fast Collision Resolution (FS-FCR) Algorithm

Fair scheduling issues in wireless local area networks have distinct characteristics from those in wired networks, so, some of the fair scheduling algorithms developed in wired networks cannot be used directly in wireless networks. Therefore, fair scheduling algorithms under distributed MAC protocols for wireless LANs are highly desirable. In the IEEE 802.11 MAC, since a station will have the initial (minimum) contention window size after its successful transmission, it will have high probability to gain access of the channel again in the next contention cycle, thus, the IEEE 802.11 MAC has inherent unfairness characteristics. Many researchers have pointed out the fairness problem in the IEEE 802.11 MAC protocol[62, 89, 98]. The fast collision resolution (FCR) algorithm also has the same problem because it follows this same procedure. In fact, it is even worse because the FCR also increases the contention window sizes of the deferring stations. To improve the fairness performance of the FCR algorithm, we propose to combine the self-clocked fair queueing (SCFQ) algorithm[43, 98] with the FCR algorithm. Our goal is to maintain the high throughput performance of the FCR algorithm while providing the high degree of fairness by utilizing the SCFQ scheduling algorithm.

3.1 Fair Scheduling Algorithms

In this section, we review the idealized GPS algorithm and the self-clocked fair queueing (SCFQ) algorithm, one of the popular packet-based GPS approximation algorithms. The distributed SCFQ algorithm is followed. We then present the proposed fairly scheduled fast collision resolution (FS-FCR) algorithm, which

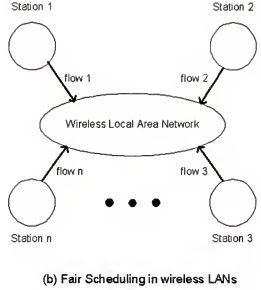
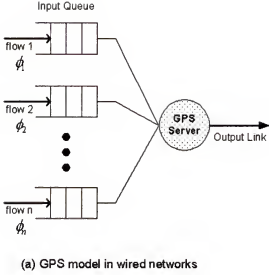


Figure 3.1: Fair scheduling system model

incorporates the distributed self-clocked fair queueing (SCFQ) algorithm into the fast collision resolution (FCR) algorithm to provide a high degree of fairness while preserving the high throughput performance.

3.1.1 Generalized Processor Sharing (GPS)

Consider the GPS system shown in Figure 3.1 (a), where the GPS server maintains n queues, which store the traffic to be served on an output link with the capacity R . A fair scheduling algorithm is used to determine which flow to be served so that a certain fairness criterion can be met. A GPS server that serves n flows is characterized by the relative amount of service for each flow (i.e., relative weight for each flow), $\phi_1, \phi_2, \dots, \phi_n$. Let $W_i(t_1, t_2)$ denote the amount of flow i traffic served in the interval $[t_1, t_2]$. If flow i does not receive any service during $[t_1, t_2]$, then $W_i(t_1, t_2) = 0$. The flow is backlogged at time t if a positive amount of that flow's traffic is queued. Then, a GPS server is defined as one for which for any flow i that is continuously backlogged in the interval $[t_1, t_2]$, we have:

$$\frac{W_i(t_1, t_2)}{W_j(t_1, t_2)} \geq \frac{\phi_i}{\phi_j} \quad \forall_j \quad (3.1)$$

for any time interval $[t_1, t_2]$ [83]. From (3.1) we obtain

$$W_i(t_1, t_2)\phi_j \geq \phi_i W_j(t_1, t_2).$$

Summing up over all j , we obtain

$$W_i(t_1, t_2) \sum_j \phi_j \geq \phi_i \sum_j W_j(t_1, t_2) = \phi_i(t_2 - t_1)R$$

from which we obtain the served amount for flow i traffic as

$$\frac{W_i(t_1, t_2)}{t_2 - t_1} \geq \frac{\phi_i}{\sum_j \phi_j} R,$$

which implies that flow i is guaranteed to be served an allocated rate r_i :

$$r_i = \frac{\phi_i}{\sum_j \phi_j} R \quad (3.2)$$

The equation (3.1) provides the clue how to design the fair scheduling algorithm. The GPS server ensures that all backlogged flows share the remaining bandwidth in proportion to the assigned weights (the same as max-min fairness concept). The set of parameters ϕ_i provides the flexibility of the GPS algorithm and can be adapted to various QoS requirements for various flows.

3.1.2 Self-Clocked Fair Queueing (SCFQ)

GPS is an idealized discipline with assumption that the server can serve multiple flows simultaneously and that the traffic is infinitely divisible. Therefore, GPS cannot be implemented accurately in practice because data transmission in real networks is packetized. Many researchers have proposed fair scheduling algorithms to approximate the idealized GPS algorithm under packet-based network environments[43, 46, 83]. self-clocked fair queueing (SCFQ) algorithm is one of the PFS algorithms, which provides desirable fairness performance while preserving implementation simplicity[43]. The basic idea of the SCFQ algorithm is that each packet is assigned a fairly scheduled transmission order in its service tag

when it arrives at the input queue and will be served in the increasing order of its service tag. The service tag is determined from the so-called *virtual time*, a time function associated with the corresponding fair queueing system, which represents the progress of work in the system. The virtual time $v(t)$ is defined as a function of time which changes with a rate equal to the rate of change of the total service provided to a session during a time period of length t .

The basic operations of the SCFQ algorithm are described as follows:

1. Each arriving packet is tagged with a service tag before it is placed in the queue. The packets in the queue are served in the increasing order of the associated service tags.
2. When a k -th packet of flow i , P_i^k , arrives at the input queue, its service tag F_i^k is assigned as follows:

$$F_i^k = \max\{v(a_i^k), F_i^{k-1}\} + \frac{L_i^k}{\phi_i}$$

where $v(a_i^k)$ is the virtual time at the time instance of a_i^k , a_i^k is the real time when packet P_i^k arrives, L_i^k is the size of packet P_i^k , and ϕ_i is the weight of flow i .

3. The virtual time $v(t)$ is updated whenever there is a packet transmission. The virtual time is set to the service tag of that packet transmitted. Intuitively, the virtual time $v(t)$ represents the normalized fair amount of service that each flow should have received. Once a busy period is over, i.e., when the server is idle, the virtual time is reset to zero.

3.1.3 Distributed Self-Clocked Fair Queueing

The self-clocked fair queueing (SCFQ) algorithm shows very good fairness performance in wired networks with very simple implementation. However, it seems that a centralized scheduler is still needed. To implement the SCFQ algorithm in

wireless LANs where a distributed contention-based MAC algorithm is operated, we have to consider some special characteristics of wireless local area networks.

The wireless LANs (such as IEEE 802.11 networks) shown in Figure 3.1 (b) have the following features:

1. Flow i corresponds to the station i , the service of flow i is equivalent to a successful packet transmissions by station i .
2. A fully distributed contention-based MAC protocol is used.
3. Each station independently determines when to transmit a packet in the contention-based MAC algorithm without the information of the network status (such as the number of active stations, service status of other stations and flow status). Therefore, the packet with the smallest service tag may not be guaranteed to be transmitted first.

Vaidya et al. proposed the distributed fair scheduling (DFS) protocol which is based on the IEEE 802.11 MAC and SCFQ algorithm. The DFS protocol transmits the packet whose finish tag is smallest, as well as SCFQ's mechanism for updating the virtual time. A distributed approach for determining the smallest finish tag is employed, using the backoff interval mechanism from the IEEE 802.11 MAC. The essential idea is to choose a backoff interval that is proportional to the finish tag of packet to be transmitted. The DFS algorithm is described as follows.

Each station i maintains a local virtual clock, $v_i(t)$, where $v_i(0) = 0$. Now, P_i^k represents the k -th packet arriving at the flow at station i on the LAN.

1. Each transmitted packet is tagged with its finish tag.
2. When at time t node i hears or transmits a packet with finish tag Z , node i sets its virtual clock v_i equal to $\text{maximum}(v_i(t), Z)$.
3. Start and finish tags for a packet are not calculated when the packet arrives.

Instead, the tags for a packet are calculated when the packet reaches the front of its flow. When packet P_i^k reaches the front of its flow at node i , the packet

is stamped with start tag S_i^k , calculated as, $S_i^k = v(f_i^k)$, where f_i^k denotes the real time when packet P_i^k reaches the front of the flow. Finish tag F_i^k is calculated as follows, where appropriate choice of the scaling factor allows us to choose a suitable scale for the virtual time.

$$F_i^k = S_i^k + \text{ScalingFactor} \times \frac{L_i^k}{\phi_i}$$

4. The objective of the next step is to choose a backoff interval such that a packet with smaller finish tag will ideally be assigned a smaller backoff interval. This step is performed at time f_i^k . Specifically, node i picks a backoff interval B_i for packet P_i^k , as a function of F_i^k and the current virtual time $v_i(f_i^k)$, as follows:

$B_i = \lfloor F_i^k - v(f_i^k) \rfloor$ slots. Now, observe that, since $F_i^k = v(f_i^k) + \text{ScalingFactor} * \frac{L_i^k}{\phi_i}$, the above expression reduces to:

$$B_i = \lfloor \text{ScalingFactor} \times \frac{L_i^k}{\phi_i} \rfloor \quad (3.3)$$

Finally, to reduce the possibility of collisions, the B_i is chosen as $B_i = \lfloor \rho * B_i \rfloor$, where ρ is uniformly distributed in $[0.9, 1.1]$. When this step is performed, a variable named CollisionCount is reset to 0.

5. If a collision occurs, then the following procedure is used. Let node i be one of the nodes whose transmission has collided with some other node. Node i chooses a new backoff interval as follows: (1) Increment CollisionCounter by 1; (2) Choose new B_i uniformly distributed in $[1, 2^{\text{CollisionCounter}-1} * \text{CollisionWindow}]$, where CollisionWindow is a constant parameter. If CollisionWindow is chosen to be small, the above procedure tends to choose a relatively small B_i after the first collision for a packet. The motivation for choosing small B_i after the first collision is as follows: The fact that node i was a potential winner of the contention for channel access indicates that it is node i 's turn to transmit in the near future. Therefore, B_i is chosen to be

small to increase the probability that node i wins again soon. However, to protect against the situation when too many nodes collide, the range for B_i grows exponentially with the number of consecutive collisions.

We observe that in the linear scheme, backoff interval B_i is a linear function of finish tag, and directly proportional to $(1/\text{flow weight})$. This can make the backoff intervals large, when flow weights are small.

3.2 Fairly-Scheduled Fast Collision Resolution (FS-FCR)

We use the distributed self-clocked fair queueing idea (SCFQ)[98, 43] in conjunction with the proposed fast collision resolution (FCR) algorithm to give a high degree of long-term fairness for best-effort data traffic while preserving the high throughput characteristics of the FCR. Instead of changing backoff values as in distributed SCFQ algorithm[98], we change the maximum successive transmission limit ($T_{PkTrans}$) for each station, which is determined by the difference between the current virtual clock value and the current finish tag value at the front of a flow. Since there are successive packet transmissions when each station gains the channel access in the FCR algorithm, the maximum successive transmission limit is updated to minimize the discrepancies between the virtual clock value and the current finish tag value at the front of a flow. Let $v(t)$ denote the virtual clock time at the real time instant t . The basic operations are characterized in the following few steps:

1. Each arriving packet to the queue of a station is tagged with a service tag before it is placed in the queue.
2. When a k -th packet of station i , P_i^k , arrives at the queue of the station, a service tag F_i^k is assigned as follows:

$$F_i^k = \max\{v(a_i^k), F_i^{k-1}\} + \frac{L_i^k}{\phi_i}$$

3. The virtual time $v(t)$ is updated whenever there is a successful packet transmission. The virtual time is set to the service tag of that packet just successfully transmitted. The virtual time $v(t)$ *approximately* represents the normalized fair amount of packet transmissions that each station should have performed (because a packet with the smallest service tag shall not be guaranteed to be served first in distributed systems). Once a busy period is over, i.e., when all stations do not have any packets to transmit, the virtual time is reset to zero.
4. Whenever a new station acquires the medium for packet transmissions, the maximum successive transmission limit (i.e., the successive transmission time period) of the station i , $T_{PkTrans,i}$, is determined by the difference between the virtual time $v(t)$ and the service tag F_i^k at the front of the packet flow at station i . If the service tag of station i is much smaller than the current virtual time, then its maximum successive transmission limit is assigned large enough to reduce the discrepancy between the current virtual time and the service tag at the front of flow i . If the service tag of station i is close to or larger than the current virtual time, then its maximum successive transmission limit is assigned to the minimum or small value to avoid increasing the discrepancy between the current virtual time and the service tag at the front of the packet flow of station i .

An example for assigning the maximum successive transmission limit is:

$$T_{PkTrans,i} = g[v(t) - F_i^k]$$

where

$$g[x] = \begin{cases} 20 \times t_s, & x \leq (-1000 \times t_s) \\ 40 \times t_s, & (-1000 \times t_s) < x \leq (-500 \times t_s) \\ 60 \times t_s, & (-500 \times t_s) < x \leq (0 \times t_s) \\ 400 \times t_s, & (0 \times t_s) < x \leq (500 \times t_s) \\ 1000 \times t_s, & (500 \times t_s) < x \leq (1000 \times t_s) \\ 2000 \times t_s, & (1000 \times t_s) < x \leq (2000 \times t_s) \\ 3000 \times t_s, & (2000 \times t_s) < x \leq (3000 \times t_s) \\ 4000 \times t_s, & (3000 \times t_s) < x \leq (4000 \times t_s) \\ 5000 \times t_s, & x > (4000 \times t_s) \end{cases}$$

where t_s is the *aSlotTime*.

5. Use the same operations of the FCR algorithm, except that, if a station reaches its maximum successive transmission limit in its packet transmission period, the station will set its contention window size to the maximum value of *maxCW*. This will give other stations higher probabilities to transmit their packets at next contention period. Since the wasted idle slots are limited less than 18, the overheads caused by the idle backoff slots will be small even after a station has finished its packet transmission period and does not have any packets to transmit.

In the FS-FCR algorithm, the SCFQ algorithm is modified to incorporate the good operational features of the fast collision resolution (FCR) MAC algorithm. Therefore, we can combine two algorithms and control the successive transmission period of the FCR algorithm by using the distributed SCFQ algorithm. In this way, we can achieve high throughput and high degree of fairness simultaneously. We will demonstrate this point via extensive simulation studies next.

3.3 Performance Evaluation

In the FS-FCR algorithm, the maximum transmission period ($T_{PkTrans}$) is controlled by the modified SCFQ algorithm to provide a high degree of fairness. Figures 3.2 and 3.3 show the results of the fairness index of FS-FCR, FCR, and IEEE802.11 MAC algorithms. The average transmission time for a packet (i.e., the

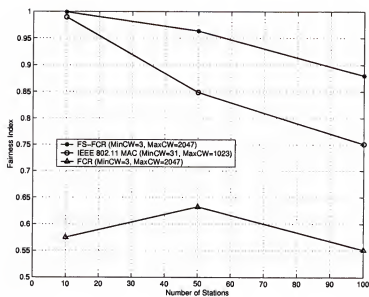


Figure 3.2: Fair index for 10 sec simulation

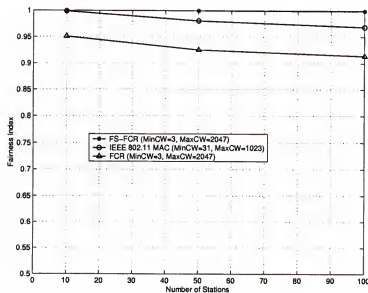


Figure 3.3: Fair index for 100 sec simulation

average packet length) of 2000 μsec (500 bytes) is used and the simulations are run for 10 and 100 seconds. We use the fairness index defined by Jain[54] to evaluate the degree of fairness for each algorithm. This fairness index is defined as

$$FairnessIndex = \frac{(\sum_i T_i/\phi_i)^2}{n \cdot \sum_i (T_i/\phi_i)^2} \quad (3.4)$$

where n is the number of flows, T_i is the throughput of flow i , ϕ_i is the weight of the flow i (we assume all stations have the same weight in simulations). From Cauchy-Schwartz inequality, we obtain $FairnessIndex \leq 1$, the equality holds if and only if all T_i/ϕ_i ($i = 1, 2, \dots, n$) are equal. Thus, the intuition behind this index is that the higher the fairness index (i.e., closer to 1), the better in terms of fairness.

From Figures 3.2 and 3.3 , we observe that the FS-FCR shows the best fairness performance, and the original FCR algorithm shows the poorest short-term fairness performance. For 10 seconds simulations, the FS-FCR algorithm provides a high degree of fairness for 10 stations wireless LANs, and the fairness index is degraded slightly as the number of stations increases to 50 and 100. The FCR algorithm shows poor fairness performance, which is worse than the IEEE 802.11 MAC algorithm in the 10 seconds simulations as we expect. In Figure 3.3, the fairness performance results of all algorithms are improved because the simulation time is long enough (100 seconds) to give sufficient opportunities for all stations to transmit, which implies that both IEEE 802.11 MAC and FCR algorithm can give long-term fairness as we can expect, while the FS-FCR algorithm addresses both short-term and long-term fairness. According to the simulation results for fairness index, we can conclude that the FS-FCR algorithm significantly improves the fairness performance of the IEEE 802.11 MAC and the FCR algorithm.

Figures 3.4 and 3.5 present the throughput results for the three protocols (FCR, FS-FCR, IEEE 802.11 MAC) for 10 and 100 contending stations wireless

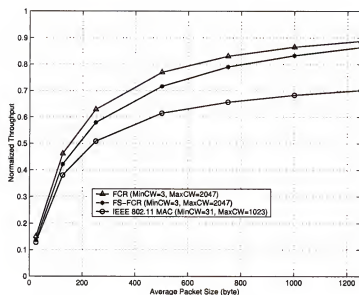


Figure 3.4: Throughput for 10 BE data station wireless LAN

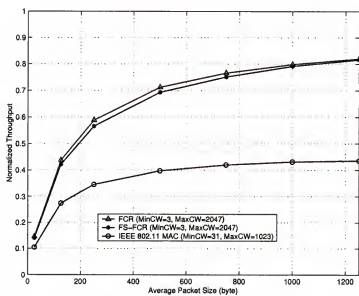


Figure 3.5: Throughput for 100 BE data stations wireless LAN

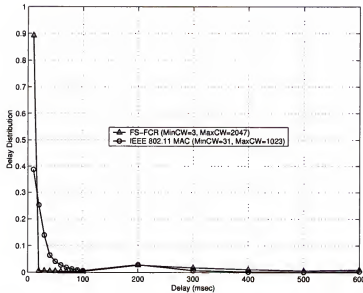


Figure 3.6: Delay Distribution for 10 stations wireless LAN

LANs, where the average transmission time for a packet (i.e., the average packet length) changes from $100 \mu\text{sec}$ (25 bytes) to $5000 \mu\text{sec}$ (1250 bytes), and the simulation time is 100 seconds. We can see that the consideration of the distributed SCFQ algorithm into the FCR algorithm causes only slight throughput degradation. Figure 3.4 shows that the throughput for the FS-FCR algorithm is much higher than that for the IEEE 802.11 MAC algorithm, and yet it is pretty close to that for the FCR algorithm. This is much more evident for 100 stations case as shown in Figure 3.5.

Figures 3.6 and 3.7 show the packet delay distribution for the IEEE 802.11 MAC algorithm and the FS-FCR algorithm for 10 and 100 stations wireless LANs with the average transmission time for a packet (i.e., the average packet length) of $2000 \mu\text{sec}$ (500 bytes). In Figure 3.6, the FS-FCR algorithm delivers 90% of all packets within 10 msec and the remaining 10% packets are delivered in the delay range from 10 msec to over 600 msec. However, the IEEE 802.11 MAC delivers only 39% packets within 10 msec, 25% packets in the range of [10 msec - 20 msec], 13% packets in the range of [20 msec - 30 msec]. In Figure 3.7, the

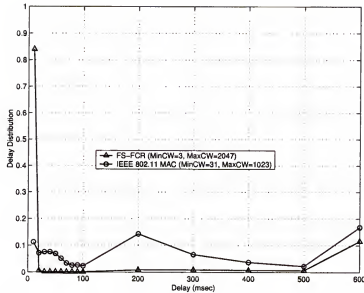


Figure 3.7: Delay Distribution for 100 stations wireless LAN

FS-FCR algorithm delivers 85% of all packets within 10 msec packet delay, while the IEEE 802.11 MAC has much longer packet delay.

From the simulation study, we observe that the FS-FCR algorithm achieves much higher degree of fairness than the IEEE 802.11 MAC algorithm while still preserving the high throughput close to that for the FCR algorithm. The good performance of the FS-FCR algorithm comes from the following factors: efficient collision resolution algorithm of the FCR algorithm and fair scheduling algorithm (SCFQ algorithm) implementation.

CHAPTER 4

QoS-based Medium Access Control Protocols

Generally, QoS refers to the concept of being able to handle a variety of desired data transmission rates, or throughput, error rates, and delay constrained traffic. To support QoS in wireless networks, all upper layer QoS components (QoS routing, QoS signaling, admission control policies, etc.) are dependent on and coordinate with the QoS support MAC protocols. However, it should be noted that there is currently no agreement as to even what a QoS-based MAC policy should look like for the current generation of radios. In the last years, considerable effort has been made to provide QoS to wireless networks with two principal approaches: reservation-based MAC protocols and contention-based MAC protocols. Reservation-based MAC protocols, such as those operating on polling, dynamic resource reservation, distributed reservation algorithms, are proposed to support the guaranteed QoS for real-time traffic with a high degree of flexibility in providing services and working efficiently under heavy network load. However, it is well known that they suffer from complex system architectures, huge overheads under low network loads and various user populations, and long restoration procedure under severe noise conditions. Moreover, reservation-based MAC protocols are usually based on sophisticated admission control and resource reservation mechanisms in order to provide guarantees or statistical assurances for absolute performance measures, such as minimum service rate or maximum end-to-end delay.

Recently, there have been some new proposals for supporting QoS in MAC layer that are based on relative performance levels instead of absolute ones. They

are mainly focused on the priority schemes in contention-based MAC protocols. Due to the nature of these schemes, admission control can be omitted, and the desired relative performance for QoS can be achieved independently in contention-based MAC protocols. The absolute performance model of reservation-based MAC protocols is well suited for real-time traffic, which require a specific capacity. However, the requirements of elastic real-time traffic can be handled better by a relative performance model in contention-based MAC protocols. Elastic applications do not require a specific capacity, instead, they use as much capacity as possible, and can still work well at low data rates. It is very challenging to address QoS issues in the distributed contention-based MAC protocols with priority schemes because there is a huge potential for these implemented easily into most of current wireless networking products which are using the contention-based MAC protocol, carrier sense multiple access/collision avoidance (CSMA/CA).

Therefore, we propose a priority scheme to support the QoS for real-time traffic. Though it does not guarantee the desired QoS, it does satisfy the QoS requirements for real-time applications. We extend the FCR algorithm by incorporating the priority algorithm based on service differentiations[1, 29] to support QoS for real-time and data services. The prioritized FCR algorithm can achieve the high throughput for best-effort data traffic transmissions while at the same time supporting QoS for real-time applications.

4.1 Current QoS-based MAC Algorithms for Real-Time Services

Many research groups are studying the QoS issues based on CSMA protocols for wireless networks. However, this is a challenging task in distributed contention-based MAC protocols because of the inherent random access characteristic[8, 99]. A similar effort has been undertaken in the IEEE 802.11 Task Group E, which has concentrated on developing QoS capabilities to the MAC algorithms for real-time traffic[52]. To provide the required QoS for real-time traffic in the presence of

data traffic in a distributed contention-based MAC, many methods have been proposed. For instance, special beacon signals may be used to notify all stations of the existence of real-time traffic transmissions, or different interframe spacing (IFS) may be used to assign a high priority to the real-time traffic. Other techniques include changing the contention window sizes and backoff ranges according to traffic types in service, or operating in two different modes such as the PCF and the DCF modes in the IEEE802.11 standard[8, 18, 33, 52, 99].

4.1.1 Differentiation Mechanism for IEEE 802.11

In order to provide either statistical or relative QoS guarantees, a service differentiation mechanism is proposed by considering different QoS parameters. There are two general approaches: using different interframe spaces (IFSs) and using different random backoff time ranges. The basic idea of using different IFSs is that if the shorter IFS a station uses, the higher priority this station will get. Since only two kinds of IFSs, SIFS and DIFS, are used in the IEEE 802.11 DCF protocol and the other IFS, PIFS, is shorter than DIFS but longer than SIFS. Therefore, PIFS has higher priority for accessing the medium than DIFS. The total time a station waits (IFS + random backoff time) should be carefully determined because no matter how short IFS a station uses, it can still lose in the contention if the total time is longer than the other station's interframe space.

In CSMA/CA, for each retransmission attempt, the backoff time grows as $\lfloor \text{rand}() \times 2^{2+i} \rfloor \times \text{aSlotTime}$, where i is the number of consecutive times a station attempts to send a frame, $\text{rand}()$ is a random number of the uniform distribution in $[0,1]$, and $\lfloor x \rfloor$ represents the largest integer less than or equal to x . To support priority by using different backoff ranges, the backoff time generation function can be changed to $\lfloor \text{rand}() \times 2^{2+i}/2 \rfloor$ for high priority stations and $2^{2+i}/2 + \lfloor \text{rand}() \times 2^{2+i}/2 \rfloor$ for low priority stations. This technique divides the random backoff time into two parts: $[0, 2^{2+i}/2 - 1]$ and $[2^{2+i}/2, 2^{2+i} - 1]$. The high

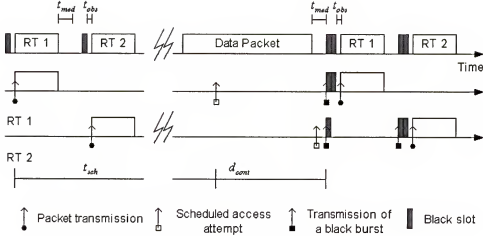


Figure 4.1: BB Contention Channel Access Scheme

priority stations use the former and the low priority stations use the latter. This is a simple example and a lot of variations have been proposed for priority based QoS support MAC algorithms[1, 29].

4.1.2 Black Burst Contention

Sobrinho et al.[91] proposed the *Black Burst* (BB) contention algorithm which provides QoS guarantee for real-time traffic. They obtained the bounded end-to-end delay at the MAC layer. In the BB contention algorithm, real-time stations contend for accessing the channel with shorter interframe space than best-effort data traffic stations, thus real-time traffic stations as a group have higher priority over data traffic stations. For channel access, real-time stations transmit Black Burst signals to set their access rights according to their waiting times. In BB contention algorithm, if the first real-time packet of a session is successfully transmitted, the succeeding real-time packets are transmitted without any collisions. BB signals are formed by an integral number of black signals and the length b of the BB sent by the station is a direct function of the contention delay d_{cont} .

BB contention supports a bounded end-to-end delay, which implies a bounded packet delay at the MAC layer. Real-time nodes contend for access to the channel after a medium interframe spacing of length t_{med} , rather than after the long interframe spacing of length t_{long} , used by data nodes. Thus, real-time nodes as a group have priority over data nodes. Instead of sending their packets when the channel becomes idle for t_{med} , real-time nodes first sort their access rights by jamming the channel with pulses of energy, denominated BBs. The length of a BB transmitted by a real-time node is an increasing function of the contention delay experienced by the node, measured from the instant when an attempt to access the channel has been scheduled until the channel becomes idle for t_{med} , i.e., until the node starts the transmission of its BB.

A real-time node uses CSMA/CA to convey its first packet until it is successful. Packet transmissions continue until the session is over. Whenever a real-time node transmits a packet, it further schedules its next transmission attempt to a time t_{sch} in the future, where t_{sch} is the same for all nodes. The length b of the BB sent by the node is a direct function of the contention delay it incurred, d_{cont}

$$b(d_{cont}) = (1 + \lfloor \frac{d_{cont}}{t_{unit}} \rfloor) t_{bslot} \quad (4.1)$$

where t_{bslot} is the length of a black slot, t_{unit} is the unit of time used to convert contention delays into an integral number of black slots, $\lfloor x \rfloor$ is the floor of x , i.e., the largest integer not larger than x .

The start of packet transmissions from different nodes are shifted in time by at least t_{pkt} . Since it is only when a node initiates the transmission of a packet that it schedules its next transmission attempt to a time t_{sch} in the future, the contention delays of different nodes will likewise differ by at least t_{pkt} . Therefore, taking $t_{unit} \leq t_{pkt}$, the BBs of different nodes differ by at least one black slot.

4.2 Real-Time Fast Collision Resolution (RT-FCR)

In order to cope with the QoS requirements of real-time applications, many algorithms have been proposed in contention-based MAC protocols for wireless LANs. The most popular approach is to use a priority scheme for each traffic type, i.e., real-time traffic has higher priority for medium access than best-effort data traffic. With higher priority for medium access, real-time traffic will be served earlier than best-effort data traffic, which results in relative performance improvements for real-time traffic over data traffic.

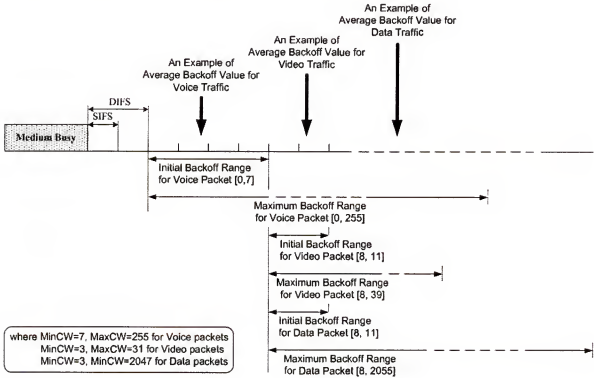


Figure 4.2: RT-FCR Medium Access Scheme

In the real-time FCR (RT-FCR) algorithm, we give priorities for accessing a medium by assigning different backoff ranges based on each of three main traffic types: voice, video, and best-effort data traffic. Intuitively, the smaller the backoff range is, the higher the priority for accessing a medium. The basic medium access scheme with three different traffic types is shown in Figure 4.2, and the backoff

ranges for the medium access are assigned according to each traffic type which is shown in Table 4.1.

Table 4.1: Assigning Backoff Range

Backoff range for voice traffic	$\{[0,7]:[0,15]:[0,31]:[0,63]:[0,127]:[0,255]\}$
Backoff range for video traffic	$\{[0,3]:[0,7]:[0,15]:[0,31]\}+8$
Backoff range for data traffic	$\{[0,3]:[0,7]:[0,15]:[0,31]:[0,63]:[0,127]:[0,255]:[0,511]:[0,1023]:[0,2047]\}+8$

In Figure 4.2 and Table 4.1, we can see that the proposed medium access algorithm effectively provides “soft” reservation to a station for the medium access according to the traffic type. In this scheme, voice traffic has the highest priority (i.e., the smallest average backoff value), and video traffic has higher priority over best-effort data traffic because of different backoff regions according to the traffic type. The access guaranteed initial backoff range $[0, 7]$ is given to voice traffic, i.e., only voice packets can be transmitted on this backoff range and other packets (video or data) will be transmitted beyond this backoff range which is shown in the backoff ranges for video and data traffic in Table 4.1 (for these backoff ranges, the constant 8 is added to move the backoff ranges for video and data traffic beyond the initial backoff range of voice traffic). Video traffic uses a much smaller maximum contention window size than best-effort data traffic in order to give higher priority over best-effort data traffic for the medium access, i.e., video traffic will have a smaller average backoff value than data traffic which is shown in Figure 4.2.

In addition to assigning different backoff ranges, the RT-FCR algorithm uses different contention algorithms with considering each traffic type. The basic procedures for the priority scheme of the RT-FCR algorithm are shown in Figure 4.3 and explained as follows:

1. *Voice Packet*: IEEE 802.11 MAC algorithm with the minimum contention window size of 7 and the maximum contention window size of 255 is used for a station with voice traffic. It has the access guaranteed initial backoff

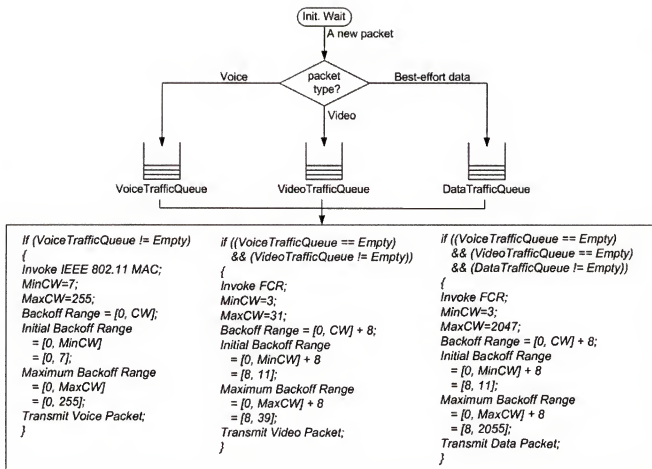


Figure 4.3: Priority Scheme of RT-FCR Algorithm

range $[0, 7]$, which gives the highest priority to voice traffic for accessing the medium. Voice traffic needs repeated packet transmissions in constant time intervals (e.g., only one packet transmission is needed every 30 ms). The FCR algorithm works with high efficiency for best-effort data traffic transmission, where each active station has more than one packets to transmit. However, in voice traffic transmissions where only one packet transmission is needed every 30 ms, the IEEE 802.11 MAC is more suitable because it does not increase the contention window sizes of the deferred stations. That is, after one station succeeds in transmitting a packet, and leaves the contention session, the remaining stations still keep the same contention window sizes and contend again (in the FCR algorithm, these remaining stations increase the contention window sizes). This results in small wasting idle slots in voice traffic transmissions.

2. *Video Packet* : Fast collision resolution (FCR) algorithm with the minimum contention window size of 3 and the maximum contention window size of 31 is used for video packet transmissions. It starts the contention for video packet transmissions after the initial backoff range of voice traffic. The smaller maximum contention window size of video traffic (MaxCW=31) than that of best-effort data traffic (MaxCW=2047) gives video traffic higher priority for the medium access over best-effort data traffic.
3. *Best-Effort Data Packet*: Fast collision resolution (FCR) algorithm with the minimum contention window size of 3 and the maximum contention window size of 2047 is used for best-effort data traffic. It starts the contention for best-effort data packet transmissions after the initial backoff range of voice traffic. FCR scheme with the large maximum contention window size achieves the high throughput for best-effort data traffic in addition to providing the opportunity for the medium access to voice or video traffic.

4.3 Performance Evaluation

In this section, we present the simulation studies for the RT-FCR algorithm in frequency hopping spread spectrum(FHSS) wireless LANs [53].

4.3.1 Source Models

We consider three different types of traffic: constant bit rate (CBR) voice traffic, variable bit rate (VBR) video traffic, and best-effort data traffic. Voice sources have two phase process with talkspurts and silent gaps. During talkspurts period, voice sources generate CBR traffic. H.263 video sources generate VBR traffic with 40 ms interframe period. We assume that best-effort data sources always have packets to transmit. The detailed source models used in our simulations are described as follows:

1. *Voice Model*[20, 47]: A voice source has two states, talkspurts and silent gaps identified by a speech activity detector. The probability that a principal talkspurt, with mean duration t_1 second, ends in a time slot of duration τ seconds is $\gamma = 1 - \exp(-\tau/t_1)$. The probability that a silent gap, of mean duration t_2 seconds, ends during τ seconds time slot is $\sigma = 1 - \exp(-\tau/t_2)$. Measured mean values for t_1 of principal talkspurts and t_2 of principal silent gaps are 1.00 and 1.35 seconds. We use 32 kbps voice traffic sources which generate one 120 byte payload voice packet every 30 msec during talkspurts period, and we assign the deadline for voice packet delay as 30 msec (i.e., the maximum voice packet delay is 30 msec).
2. *Video Model*[20, 64]: We use the H.263 video traffic with 40 msec interframe period, i.e., 25 frames per second. During an interframe period, each video source generates a frame consisting of a variable number of packets. As soon as packets become available from the coder, they could be transmitted at the maximum rate the channel allows. The video packet size is 120 bytes and the mean rate of video traffic is 48 kbps and the maximum rate is 480 kbps. That

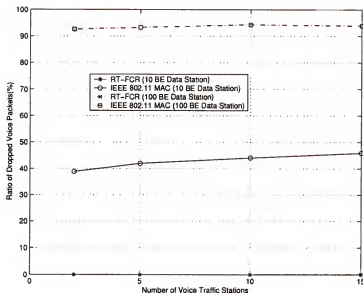


Figure 4.4: Ratio of Dropped Voice Packets

is, there are 2 packets per frame for the mean rate and the maximum number of packets per frame is 20. We use the deadline for video packet delay as 120 msec.

3. *Best-effort Data Model*[18]: It is assumed that best-effort data sources always have packets to transmit. We use the parameter $q = 0.975$ from the geometric distribution for best-effort data packet length, which implies that the average packet length of best-effort data traffic is 40 slots.

4.3.2 Simulation Results

We present the simulation results of the RT-FCR algorithm for 10 and 100 best-effort data traffic stations with varying the number of CBR voice traffic stations up to 15. We compare the results of the RT-FCR algorithm with those of the IEEE 802.11 MAC algorithm. The ratio of the dropped voice packets to the total generated voice packets is shown in Figure 4.4, and the throughput for the best-effort data traffic transmissions is shown in Figure 4.5. In Figure 4.4, the IEEE 802.11 MAC algorithm loses over 40% of voice packets with 10 best-effort data stations and over 90% with 100 best-effort data stations. This is expected

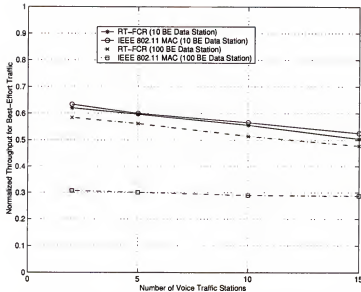


Figure 4.5: Throughput of Best-Effort Data Traffic Transmission

because the IEEE 802.11 DCF mode treats real-time traffic the same as the best-effort data traffic. The ratios of dropped voice packets for the RT-FCR algorithm are close to zero for both cases. The RT-FCR algorithm shows very low voice packet dropping ratio while still preserving the high throughput performance for best-effort data traffic, which is obvious in Figures 4.4 and 4.5.

We carry out the performance evaluation of the RT-FCR algorithm for the integration of three different traffics: voice, video, and best-effort data. Figure 4.6, 4.7, 4.8, and 4.9 show the performance results of the RT-FCR algorithm for the integration of three different traffics. The number of best-effort data stations is 10 for all simulations. Figure 4.6 shows that the ratio of the dropped real-time packets to the generated real-time packets vs. various numbers of CBR voice stations with 10 best-effort data stations and 5 VBR video stations. The throughput of the best-effort data traffic for this case is shown in Figure 4.7. In Figure 4.6 and 4.7, we can see that the RT-FCR algorithm can support the desired QoS for real-time applications upto 30 CBR stations with 10 best-effort data stations and 5 VBR stations. Figure 4.8 shows that the result of dropped real-time

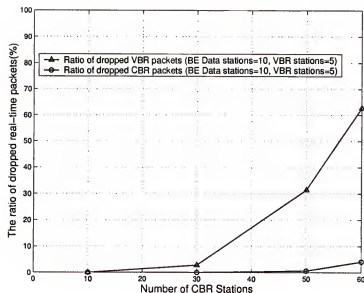


Figure 4.6: Ratio of Dropped Real-Time Packets vs. Number of CBR Stations

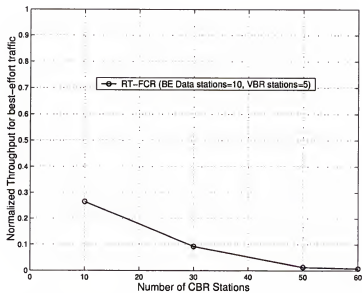


Figure 4.7: Throughput of Best-Effort Data Traffic vs. Number of CBR Stations

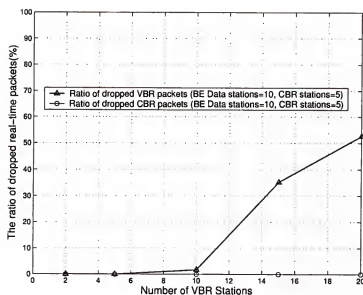


Figure 4.8: Ratio of Dropped Real-Time Packets vs. Number of VBR Stations

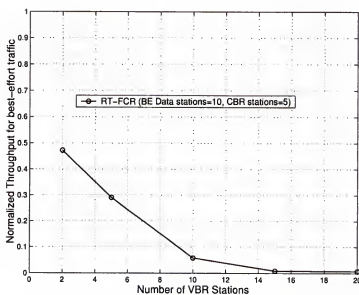


Figure 4.9: Throughput of Best-Effort Data Traffic vs. Number of VBR Stations

packets to generated real-time packets vs. various numbers of VBR video stations with 10 data stations and 5 voice stations. The throughput of best-effort data traffic for this case is shown in Figure 4.9. In Figure 4.8 and 4.9, we can see that the RT-FCR algorithm can support the desired QoS for real-time applications upto 10 VBR stations with 10 best-effort data stations and 5 voice CBR stations. Figure 4.6 and 4.8 show that voice traffic has much higher priority for channel access over video and best-effort data traffics, so the ratio of dropped packet for voice traffic is close to zero for most cases. The ratio of dropped packet for video traffic is affected by best-effort data traffic as the number of CBR stations or VBR stations increases. From the simulation results, we can conclude that the QoS for voice traffic is highly satisfied and the QoS for video traffic is satisfactory in the RT-FCR algorithm. While providing QoS for real-time traffic, the RT-FCR algorithm achieves the high throughput for best-effort data traffic when the channel is available for best-effort data traffic transmissions between real-time traffic transmissions, which is shown in Figure 4.6, 4.7, 4.8, and 4.9.

CHAPTER 5

CONCLUSIONS and Future Research Directions

Designing efficient medium access control (MAC) protocols for wireless local area networks is one of the most critical aspects of network performance and proficient system operations. Among two different research directions for MAC design in wireless LANs, i.e., reservation-based and contention-based, we focus our interest on contention-based MAC research area. Though, reservation-based MAC protocols provide guaranteed QoS support capabilities, flexibility for designing many scheduling approaches, and high efficiency under heavy network loads, the system architecture is complex, the overheads are huge under irregular data traffic, and efficient operation is in question in view of the harsh wireless channel conditions. Currently, the most pervasive MAC algorithm used in wireless LANs is the IEEE 802.11 CSMA/CA which is a distributed contention-based MAC algorithm. The IEEE 802.11 MAC algorithm works well under wireless channels (one hop situation), but it does not support QoS for real-time services nor consider fairness for serving users. In wired networks, fair scheduling or QoS support problems have been studied by many researchers. It is assumed that there is a centralized scheduler or an access point which controls or assigns the transmission orders according to the rules of providing fairness for serving users. In distributed contention-based MAC algorithms such as variants of the IEEE 802.11 CSMA/CA, there is no centralized scheduler. Therefore, each station has to operate independently without any knowledge for the network status. It is very challenging and fascinating to develop MAC protocols which provide all required characteristics such as high throughput, high degree of fairness, and the capability

of supporting QoS for real-time services under completely distributed network architectures.

First, we introduce a brief explanation for wireless local area networks both in physical layer and in MAC layer. Wireless channel characteristics and communication technology are discussed in terms of the physical layer and this is followed by a brief history of the development of contention-based MAC protocols. The important contributions of this work are divided by three categories: improving throughput, fair scheduling, and QoS support for real-time services.

We propose a new contention-based medium access control algorithms, namely, the fast collision resolution (FCR) algorithm. The FCR algorithm can achieve high throughput performance while preserving implementation simplicity in wireless local area networks. In the FCR algorithm, each station changes the contention window size upon both successful packet transmissions and collisions (i.e., upon detecting a start of busy period) for all active stations in order to redistribute the backoff timers to actively avoid potential future collisions. Due to this operation, each station can quickly resolve collisions. Other ideas we incorporate in the FCR are using smaller minimum contention window size compared with the IEEE 802.11 MAC and faster decrease of backoff timers after detecting a number of idle slots. These changes could reduce the average number of idle slots in each contention cycle, which contributes to the throughput improvement. Extensive simulation studies for throughput, delay distribution in various network configurations and transport layer performance have demonstrated that the FCR algorithm gives significant performance improvement over the IEEE 802.11 MAC algorithm.

We extend the fast collision resolution (FCR) algorithm to improve the fairness for serving users and to provide QoS for real-time applications. For the fair scheduling scheme of the FCR algorithm, we use the distributed self-clocked fair queueing algorithm, while the priority scheme based on service differentiations

is modified and incorporated with the FCR algorithm to support the QoS for real-time applications. To address the fairness issue, the fairly scheduled FCR (FS-FCR) algorithm controls the successive transmission period of the FCR algorithm to approximate the fluid-flow GPS algorithm in the long term. Extensive simulation studies for throughput, delay distribution and fairness have demonstrated that the FS-FCR algorithm achieves both high throughput and a high-degree of fairness simultaneously. In real-time FCR (RT-FCR) algorithm, the QoS for real-time and data services is supported while preserving the high throughput performance of the FCR algorithm. The priority scheme based on service differentiations is combined with the FCR algorithm to support QoS for real-time applications. Extensive simulation studies for throughput and ratio of dropped real-time packets show that the RT-FCR algorithm support efficiently the desired QoS for real-time services while providing the high throughput performance for best-effort data traffic.

Our work focuses on important topics of MAC protocol designs in wireless LANs such as throughput, fairness, and QoS support for real-time services. Our future research is to extend the RT-FCR algorithm to handle more complicated mixture of real-time and non-real-time traffics in different network configurations. The multi-layer optimization is another important topic which should be further investigated. For instance, the interaction between the MAC layer and the transport layer or the network layer may significantly affect the performance of the wireless LANs. The operational characteristics of MAC protocols in multi-hop ad hoc networks should be explored to provide good solutions for the problems caused by multi-hop nature of the wireless networks environments.

REFERENCES

- [1] I. Aad and C. Catelluccia, "Differentiation Mechanism for IEEE 802.11," *Proc. IEEE INFOCOM'01*, Anchorage, Alaska, Apr., 2001.
- [2] A. S. Acampora and S. V. Krishnamurthy, "A New Adaptive MAC Layer Protocol for Broadband Packet Wireless Networks in Harsh Fading and Interference Environments," *IEEE/ACM Trans. on Networking*, vol. 8, NO. 3, Jun., 2000.
- [3] L. Alonso, R. Agusti, and O. Sallent, "A Near-Optimum MAC Protocol Based on the Distributed Queueing Random Access Protocol (DQRAP) for a CDMA Mobile Communication System," *IEEE JSAC*, vol. 18, NO. 9, pp.1701-1718, Sep., 2000.
- [4] M. Alasti and N. Farvardin, "SEAMA: A Source Encoding Assisted Multiple Access Protocol for Wireless Communications," *IEEE JSAC*, vol. 18, NO. 9, pp.1682-1700, 2000.
- [5] G. Anastasi, L. Lenzini, and E. Mingozzi, "Stability and Performance Analysis HIPERLAN," *Proc. IEEE INFOCOM'98*, pp.134-141, 1998.
- [6] A. Baiocchi, F. Cuomo, and S. Bolognesi, "IP QoS Delivery in a Broadband Wireless Local Loop: MAC Protocol Definition and Performance Evaluation," *IEEE JSAC*, vol. 18, NO. 9, pp.1608-1622, Sep., 2000.
- [7] K. Balachandran, S. Nanda, and S. Vitebsky, "Design of a Medium Access Control Feedback Mechanism for Cellular TDMA Packet Data Systems," *IEEE JSAC*, vol. 18, NO. 9, pp.1719-1730, 2000.
- [8] A. Banchs, X. Perez, M. Radimirsch, and H. J. Stuttgen, "Service Differentiation Extensions for Elastic and Real-Time Traffic in 802.11 Wireless LAN," *IEEE Workshop on High Performance Switching and Routing*, pp.245-249, 2001.
- [9] L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla, "GloMoSim: A Scalable Network Simulation Environment," UCLA Computer Science Department Technical Report 990027, May 1999.
- [10] D. Bertsekas and R. Gallager, *Data Networks*, 2nd Ed., Prentice Hall, Englewood Cliffs, New Jersey, 1992.

- [11] B. Bensaou, Y. Wang, and C. C. Ko, "Fair Medium Access in 802.11 Based Wireless Ad Hoc Network," *IEEE/ACM Int. Workshop on Mobile Ad Hoc Networking and Computing*, Boston, MA, Aug., 2000.
- [12] V. Bharghavan, "MACAW: A Media Access Protocol for Wireless LAN's," *SIGCOMM'94*, pp.212-225, London, England, Aug. 1994.
- [13] V. Bharghavan, "Performance Evaluation of Algorithms for Wireless Medium Access," *IEEE International Computer Performance and Dependability Symposium IPDS'98*, pp.142-149, 1998.
- [14] G. Bianchi, "Performance Analysis of the IEEE802.11 Distributed Coordination Function," *IEEE Journal on Selected Areas in Communications*, Vol.18, No.3, PP.535-547 Mar. 2000.
- [15] G. Bianchi, L. Fratta, and M. Oliveri, "Performance Evaluation and Enhancement of the CSMA/CA MAC Protocol for 802.11 Wireless LANs," *Proc. of PIMRC*, Taipei, Taiwan, Oct., 1996.
- [16] H. Bolcskel, A. J. Paulray, K. V. S. Hari, R. U. Nabar, and W. W. Lu, "Fixed Broadband Wireless Access: State of The Art, Challenges, and Future Directions," *IEEE Comm. Magazine*, vol. 39, Issue 1, Jan., 2001.
- [17] D. Bushmitch, S. Mukherjee, S. Narayanan, and M. Ratty, "Supporting MPEG Video Transport on DOCSIS-Compliant Cable Network," *IEEE JSAC*, vol. 18, NO. 9, pp.1581-1596, Sep., 2000.
- [18] F. Cali, M. Conti, and E. Gregori, "Dynamain Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit," *IEEE/ACM Trans. on Networking*, vol. 8, NO. 6, pp.785-799, Dec. 2000.
- [19] F. Cali, M. Conti, and E. Gregori, "IEEE 802.11 Protocol: Design and Performance Evaluation of an Adaptive Backoff Mechanism," *IEEE JSAC*, vol. 18, NO. 9, pp.1774-1786, Sep., 2000.
- [20] J. Chen, K. M. Sivalingam, P. Agrawal, and R. Acharya, "Scheduling Multimedia Services in a Low-Power MAC for Wireless and Mobile ATM Networks," *IEEE Trans. on Multimedia*, Vol.1, NO.2, pp.187-201, June 1999.
- [21] J. Charzinski, "Activity Polling and Activity Contention in Media Access Control Protocols," *IEEE JSAC*, vol. 18, NO. 9, pp.128-133, Sep., 2000.
- [22] S. Chevrel, A. H. Aghvami, H. Y. Lach, and L. Taylor, "Analysis and Optimisation of the HIPERLAN Channel Access Contention Scheme," *Wireless Personal Communications*, vol. 4, pp.27-39, 1996.
- [23] A. Chandra, V. Gummalla, and J. O. Limb, "Wireless Medium Access Control Protocols," *IEEE Communications Surveys*, Second Quarter 2000.

- [24] S. Choi and K. Shin, "A class of Adaptive Hybrid ARQ Schemes for Wireless Links," *IEEE Trans. Vehicle Technology*, vol. 50, Issue 3, May 2001.
- [25] J. Chen, K. M. Sivalingam, P. Agrawal, and R. Acharya, "Scheduling Multimedia Services in a Low-Power MAC for Wireless and Mobile ATM Networks," *IEEE Trans. on Multimedia*, vol. 1, NO. 2, Jun., 1999.
- [26] A. Charoenpanitkit, N. Sivamok, and L. Wuttisittikulkij, "An Analysis of a New Access Control Technique for Channel Request in Wireless Communications," *IEEE ISCAS 2001*, vol. 2, pp.333-336, 2001.
- [27] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai, "IEEE 802.11 Wireless Local Area Networks," *IEEE Communications Magazine* Vol.35, pp.116-126, Sep. 1997.
- [28] F. Cuomo, A. Baiocchi, and R. Cautelier, "A MAC Protocol for a Wireless LAN Based on OFDM-CDMA," *IEEE Comm. Magazine*, pp.152-159, Sep., 2000.
- [29] J. Deng and R. S. Chang, "A Priority Scheme for IEEE 802.11 DCF Access Method," *IEICE Trans. Commun.*, Vol.E82-B, NO.1, Jan. 1999.
- [30] J. Deng and R. S. Chang, "A Nonpreemptive Priority-Based Access Control Scheme for Broadband Ad Hoc Wireless ATM Local Area Networks," *IEEE JSAC*, vol. 18, NO. 9, Sep., 2000.
- [31] A. Demers, S. Keshav, and S. Shenker, "Analysis and Simulation of a Fair Queueing Algorithm," *Proc. ACM SigComm'89*, pp.1-12, 1989.
- [32] D. A. Eckhardt and P. Steenkiste, "Effort Limited Fair (ELF) Scheduling for Wireless Networks," *IEEE INFOCOM'00*, pp.1097-1106, 2000.
- [33] *HIPERLAN Type 2 Standard*, European Telecommunications Standards Institute (ETSI), 2000.
- [34] A. Farago, A. D. Myers, V. R. Syrotiuk, and G. V. Zaruba, "Meta-MAC Protocols: Automatic Combination of MAC Protocols to Optimize Performance for Unknown Conditions," *IEEE JSAC*, vol. 19, NO. 9, Sep., 2000.
- [35] R. Fantacci and S. Nannicini, "Performance Evaluation of a Reservation TDMA Protocol for Voice/Data Transmission in Personal Communication Networks with Nonindependent Channel Errors," *IEEE JSAC*, vol. 18, NO. 9, pp.1636-1646, 2000.
- [36] C. H. Foh and M. Zukerman, "CSMA with Reservations by Interruptions (CSMA/RI): A Novel Approach to Reduce Collisions in CSMA/CD," *IEEE JSAC*, vol. 18, NO. 9, pp.1572-1580, Sep., 2000.

- [37] J. Frigon, V. C. M. Leung, and H. C. B. Chan, "Dynamic Reservation TDMA Protocol for Wireless ATM Network," *IEEE JSAC*, vol. 19, NO. 2, Feb., 2001.
- [38] C. Fullmer and J. Garcia-Luna-Aceves, "Floor Acquisition Multiple Access (FAMA) for Packet-Radio Networks," *Proc. SIGCOMM'95*, pp.262-273, Cambridge, MA.
- [39] K. Fu, Y. J. Guo, and S. K. Barton, "Performance of the EY-NPMA," *Wireless Personal Communications*, 4: pp.41-50, 1996.
- [40] R. Garces and J. Garcia-Luna-Aceves, "Collision Avoidance and Resolution Multiple Access with Transmission Groups," *Proc. IEEE INFOCOM'97*, Kobe, Japan, pp. 134-142.
- [41] J. Garcia-Luna-Aceves, A. Tzamaloukas, "Reversing the Collision-Avoidance Handshake in Wireless Networks," *ACM Mobicom'99*, pp.120-131, 1999.
- [42] J. Geier, *Wireless LANs*, SAMS, 1999.
- [43] S. J. Golestani, "A Self-Clocked Fair Queueing Schemes for Broadband Applications," *IEEE INFOCOM'94*, pp.636-646, Apr., 1994.
- [44] N. Golmie, Y. Saintillan, and D. H. Su, "A Review of Contention Resolution Algorithms for IEEE 802.14 Networks," *IEEE Comm. Survey*, pp.2-12, 1st Quarter, 1999.
- [45] D. J. Goodman, R. A. Valenzuela, K. T. Gayliard, and B. Ramamurthi, "Packet Reservation Multiple Access for Local Wireless Communications," *IEEE Transactions on Communications*, vol.37, NO.8, pp.885-890, Aug. 1989.
- [46] P. Goyal, H. M. Vin, and H. Cheng, "Start-Time Fair Queueing: A Scheduling Algorithm for Integrated Services Packet Switching Networks," *IEEE/ACM Transactions on Networking*, vol. 5, NO. 5, Oct., 1997.
- [47] D. J. Goodman and S. X. Wei, "Efficiency of Packet Reservation Multiple Access," *IEEE Trans. on Vehicular Technology*, Vol.40, NO.1, pp.170-176, Feb. 1991.
- [48] A. C. V. Gummalla and J. O. Limb, "Design of an Access Mechanism for a High Speed Distributed Wireless LAN," *IEEE JSAC*, vol. 18, NO. 9, Sep., 2000.
- [49] F. Halsall, *Data Communications, Computer Networks and Open Systems*, 4th Ed., Addison-Wesley.
- [50] D. P. Heyman, T. V. Lakshman, A. Tabatabai, and H. Heeke, "Modeling Teleconference Traffic from VBR Video Coders," *Int. Conf. on Comm. '94*, vol. 3, pp.1744-1748, 1998.

- [51] D. P. Heyman, A. Tabatabai, and T. V. Lakshman, "Statistical Analysis and Simulation Study of Video Teleconference Traffic in ATM Networks," *IEEE Trans. on Circuit and Systems for Video Technology*, vol. 2, NO. 1, Mar., 1992.
- [52] IEEE 802.11 Status of Project IEEE 802.11e,
http://grouper.ieee.org/groups/802/11/Reports/tge_update.htm
- [53] *IEEE 802.11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*, IEEE, 1997.
- [54] R. Jain, A. Duresi, and G. Babic, "Throughput Fairness Index: An Explanation," *ATM Forum/99-0045*, Feb. 1999.
- [55] M. Johnson, "HiperLAN/2-The Broadband Radio Transmission Technology Operating in the 5 GHz Frequency Band," HiperLAN/2 Global Forum, 1999.
- [56] C. Kang, C. Ahn, and W. Kang, "Contention-Free Distributed Dynamic Reservation MAC Protocol with Deterministic Scheduling for Wireless ATM Networks," *IEEE JSAC*, vol. 18, NO. 9, pp.1623-1635, 2000.
- [57] S. Kandukuri and N. Bambos, "Multimodal Dynamic Multiple Access (MDMA) in Wireless Packet Network," *IEEE INFOCOM'01*, pp.199-208, 2001.
- [58] P. Karn, "MACA-A New Channel Access Method for Packet Radio," *Proc. ARRL/CRRL Amateur Radio Ninth Computer Networking Conf.*, ARRL, 1990.
- [59] M. Kawagishi, S. Sampei, and N. Morinaga, "A Novel Reservation TDMA Based Multiple Access Scheme using Adaptive Modulation for Multimedia Wireless Communication Systems," *IEEE VTC'98*, pp.112-116, 1998.
- [60] K. Kim, S. Shin, and K. Kim, "A Novel MAC Scheme for Prioritized Services in IEEE 802.11a Wireless LAN," *ATM (ICATM 2001) and High Speed Intelligent Internet Symposium, Joint 4th IEEE International Conference*, pp.196-199, 2001.
- [61] J. G. Kim and I. Widjaja, "PRMA/DA: A New Media Access Control Protocol for Wireless ATM," *Conf. Record of ICC'96*, pp.240-244, Dallas, TX, Jun., 1996.
- [62] C. E. Koksal, H. Kassab, and H. Balakrishnan, "An Analysis of Short-Term Fairness in Wireless Media Access Protocols," *ACM SIGMETRICS 2000*, Santa Clara, CA, Jun., 2000.
- [63] H. Koraitim and S. Tohme, "Performance Analysis of Multiple Access Protocols for Multimedia Satellite Networks," *IEEE JSAC*, vol. 18, NO. 9, pp.1751-1763, Sep., 2000.

- [64] Y. Kwok and V. K. N. Lau, "A Quantitative Comparison of Multiple Access Control Protocols for Wireless ATM," *IEEE Trans. on Vehicular Technology*, Vol.50, NO.3, pp.796-815, May, 2001.
- [65] T. V. Lakshman, A. Ortega, and A. R. Reibman, "VBR Video: Tradeoffs and Potentials," *Proc. of the IEEE*, vol. 86, NO. 5, May, 1998.
- [66] L. Lenzini, M. Luise, and R. Reggiannini, "CRDA: A Collision Resolution and Dynamic Allocation MAC Protocol to Integrate Data and Voice in Wireless Networks," *IEEE JSAC*, vol. 19, Issue 6, Jun., 2001.
- [67] P. Lin, B. Bensaou, Q. L. Ding, and K. C. Chua, "Cs-WFQ: A Wireless Fair Scheduling Algorithm for Error Prone Wireless Channels," *ACM Int. Workshop on Wireless Mobile Multimedia*, pp.11-20, 2000.
- [68] Y. B. Lin and I. Chlamtac, *Wireless and Mobile Network Architecture*, Wiley.
- [69] C. Lin and M. Gerla, "Asynchronous Multimedia Multihop Wireless Networks," *Proc. IEEE INFOCOM'97*, Kobe, Japan, pp.118-125.
- [70] C. Liu, Y. Ge, M. Fitz, J. hou, W. Chen, R. Jain, "OSU-MAC: A New, Real-Time Medium Access Control Protocol for Wireless WANs with Asymmetric Wireless Links," *21st International Conference on Distributed Computing Systems*, pp.537-546, 2001.
- [71] S. Lu, V. Bharghavan, and R. Srikant, "Fair Scheduling in Wireless Packet Networks," *ACM MOBICOM'98*, Oct., 1998.
- [72] H. Luo and S. Lu, "A Topology-Independent Fair Queueing Model in Ad Hoc Wireless Networks," *Proc. IEEE ICNP'00*, Osaka, Japan, Aug., 2000.
- [73] H. Luo, P. Medvedev, J. Cheng, and S. Lu, "A Self-Coordinating Approach to Distributed Fair Queueing in Ad Hoc Wireless Networks," *IEEE INFOCOM'01*, Anchorage, AK, Apr., 2001.
- [74] A. Muir and J. J. Garcia-Luna-Aceves, "Group Allocation Multiple Access in Single-Channel Wireless LANs," *Proc. Communication Networks and Distributed Systems Modeling and Simulation Conference*, Phoenix, AZ, 1997.
- [75] A. Muir and J. J. Garcia-Luna-Aceves, "Supporting Real-Time Multimedia Traffic in a Wireless LAN," *Proc. SPIE Multimedia Computing and Networking Conference*, Feb., 1997.
- [76] L. Musumeci, P. Giacomazzi, and L. Fratta, "Polling and Contention-Based Schemes for TDMA/TDD Access to Wireless ATM Networks," *IEEE JSAC*, vol. 18, NO. 9, Sep., 2000.

- [77] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan, "Achieving MAC Layer Fairness in Wireless Packet Networks," *Proc. ACM/IEEE Mobicom*, pp.87-98, Aug., 2000.
- [78] T. Nandagopal, S. Lu, and V. Bharghavan, "A Unified Architecture for the Design and Evaluation of Wireless Fair Queueing Algorithm," *Proc. Mobicom'99*, pp.132-142, Aug., 1999.
- [79] P. Narasimhan, *Quality of Services Based Bandwidth Allocation in Integrated Multiservices Wireless Networks*, Ph.D. Thesis, 1999.
- [80] R. V. Nee, and R. Prasad, *OFDM for Wireless Multimedia Communications*, Artech House Publishers, 2000.
- [81] T. S. Ng, I. Stoica, and H. Zhang, "Packet Fair Queueing Algorithms for Wireless Networks with Location-Dependent Errors," *IEEE INFOCOM'98*, Mar., 1998.
- [82] T. Ozugur, M. Naghshineh, P. Kermani, and J. A. Copelan, "Fair Media Access for Wireless LANs," *IEEE Globecom'99*, pp.570-579, 1999.
- [83] A. K. Parekh and R. G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Single-Node Case," *IEEE/ACM Trans. on Networking*, Vol. 1. No. 3, pp.344-357, June 1993.
- [84] A. K. Parekh and R. G. Gallager, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks: The Multiple Node Case," *IEEE/ACM Trans. on Networking*, Vol. 2. No. 2, pp.137-150, June 1994.
- [85] M. E. Rivero-Angeles, D. Lara-Rodriguez, and F. A. Cruz-Perez, "A New EDGE Medium Access Control Mechanism Using Adaptive Traffic Load Allotted Aloha," *IEEE VTC'01*, vol. 3, pp.1358-1362, 2001.
- [86] O. Rose and M. R. Frater, "A Comparison of Models for VBR Video Traffic Sources in B-ISDN," Research Report No. 72, Institute of Computer Science, University of Wurzburg, Oct., 1993.
- [87] H. Rohling and R. Grunheid, "Performance Comparison of Different Multiple Access Schemes for the Downlink of an OFDM Communication System," *IEEE 47th VTC*, vol. 3, pp.1365-1369, 1997.
- [88] O. Sharon and E. Altman, "An Efficient Polling MAC for Wireless LANs," *IEEE/ACM Trans. on Networking*, vol. 9, Issue 4, pp.439-451, Aug., 2001.
- [89] S. Xu and T. Saadawi, "Does the IEEE 802.11 MAC Protocol Work Well in Multihop Ad Hoc Networks?" *IEEE Communications Magazine*, June 2001.

- [90] D. J. Skellern, A. C. K. Lam, and N. Weste, "Issues in High-Speed WLANs," *Proc. ICASSP'00*, vol. 6, pp. 3698-3701, 2000.
- [91] J. L. Sobrinho and A. S. Krishnakumar, "Quality-of-Service in Ad Hoc Carrier Sense Multiple Access Wireless Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, pp.1353-1368, Aug. 1999.
- [92] W. Stallings, *Data and Computer Communications*, 5th Ed., Prentice Hall.
- [93] W. R. Stevens, *TCP/IP Illustrated*, Vol. 1, Addison Wesley, 1994.
- [94] K. Tang, M. Correa, and M. Gerla, "Effects of Ad Hoc MAC Layer Medium Access Mechanisms under TCP," *Mobile Networks and Applications*, vol. 6, pp.317-329, 2001.
- [95] A. S. Tanenbaum, *Computer Networks*, 3rd Ed., Prentice Hall.
- [96] S. Tsao, "Extending Earliest Due Date Scheduling Algorithm for Wireless Networks with Location Dependent Errors," *IEEE VTC 2000*, pp.223-228, 2000.
- [97] A. Tzamaloukas and J. J. Garcia-Luna-Aceves, "A Receiver Initiated Collision Avoidance Protocol for Multi-Channel Networks," *IEEE INFOCOM'01*, pp.189-198, 2001.
- [98] N. H. Vaidya, P. Bahl, and S. Gupta, "Distributed Fair Scheduling in a Wireless LAN," *Proc. Mobicom'2000*, Boston, MA, USA, Aug. 2000.
- [99] M. Veeraraghavan, N. Cocker, and T. Moors, "Support of Voice Services in IEEE 802.11 Wireless LANs," *Proc. of IEEE INFOCOM'2001*, pp.488-497, Vol.1, 2001.
- [100] E. K. Wesel, *Wireless Multimedia Communications*, Addison-Wesley, 1998.
- [101] D. E. Wrege, E. W. Knightly, H. Zhang, and J. Liebeherr, "Deterministic Delay Bounds for VBR Video in Packet-Switching Network: Fundamental Limits and Practical Tradeoffs," *IEEE/ACM Trans. on Networking*, 4(3): pp.352-362, Jun., 1996.
- [102] S. Wu, Y. Tseng, and J. Seu, "Intelligent Medium Access for Mobile Ad Hoc Networks with Busy Tones and Power Control," *IEEE JSAC*, vol. 18, NO. 9, Sep., 2000.
- [103] S. Xu and T. Saadawi, "Does IEEE 802.11 MAC Protocol Work Well in Multi-hop Wireless Ad Hoc Networks?," *IEEE Communication Magazine*, Jun. 2001.
- [104] G. Xylomenos and G. C. Polyzos, "TCP and UDP Performance over a Wireless LAN," *Proc. IEEE INFOCOM'99*, pp.439-446, 1999.

- [105] W. Yin and Y. Lin, "Statistically Optimized Minislot Allocation for Initial and Collision Resolution in Hyprid Fiber Coaxial Networks," *IEEE JSAC*, vol. 18, NO. 9, pp.1764-1773, 2000.
- [106] M. C. Yuang and P. L. Tien, "Multiple Access Control with Intelligent Bandwidth Allocation for Wireless ATM Networks," *IEEE JSAC*, vol. 18, NO. 9, pp.1658-1669, 2000.

BIOGRAPHICAL SKETCH

Younggoo Kwon began his college career as an electrical engineering student at Korea University in Seoul, Korea. He received the B.S. and M.S. degrees from Korea University in 1993 and 1996, respectively, and will receive the Ph.D. degree in Aug. 2002, from the University of Florida, Gainesville, FL, all in electrical and computer engineering. His main research area is designing and developing efficient medium access control systems for multimedia applications over high-speed wireless networks. These areas include QoS guarantees for wireless network protocol design for multimedia applications; fair scheduling for high-speed wireless networks; OFDM Implementation and optimization with MAC layer in high speed Wireless LANs; powerline communication PHY and MAC layer performance analysis; medium access control for wireless ATM networks. In addition to the above, Younggoo has extensive research experience in computer network design, information theory, and cellular data network area.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Yuguang Fang, Chair
Assistant Professor of Electrical and
Computer Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Haniph Latchman, Co-chair
Associate Professor of Electrical and
Computer Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Tan Wong
Assistant Professor of Electrical and
Computer Engineering

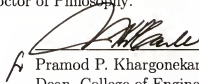
I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Zuo-Jun Shen
Assistant Professor of Industrial and
Systems Engineering

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

August 2002

A handwritten signature in black ink, appearing to read 'P. Khargonekar', is written over a horizontal line. To the left of the signature, there is a small, stylized handwritten mark that looks like a lowercase 'f' or a checkmark.

Pramod P. Khargonekar
Dean, College of Engineering

Winfred M. Phillips
Dean, Graduate School